

CRANFIELD UNIVERSITY



TOM A.I. DE VUYST

**Hydrocode Modelling of Water Impact**

SCHOOL OF ENGINEERING

PhD THESIS

ProQuest Number: 10820915

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10820915

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by Cranfield University.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code  
Microform Edition © ProQuest LLC.

ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 – 1346

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING

PhD THESIS

Academic Year 2002-3

TOM A.I. DE VUYST

**Hydrocode Modelling of Water Impact**

Supervisor: Dr. Rade Vignjevic

March 2003

## **Abstract**

This thesis addresses the problem of hydrocode modelling of water impact. Two facets that are of importance when numerically modelling the impact of metallic structures on water are metal anisotropy and water behaviour during impact. In order to be able to take account of these effects in a hydrocode simulation an SPH solver has been incorporated into LLNL-DYNA3D. The treatment of contact in meshless methods has been addressed through the development of a contact algorithm which does not require the construction of surfaces. The interaction of finite elements and SPH particles is accounted for by using a novel approach in treating the finite element nodes as particles in the contact treatment. The same contact algorithm developed for the treatment of contact in the SPH method has been used. In order to take account of metal anisotropy a material model that takes account of anisotropy in the elastic and plastic regimes, strain-rate dependency and non-linear behaviour at high pressures including spall failure was developed.

The developed simulation tool is validated against experimental data for the case of water impact of rigid cylinders on water. Further validation is achieved by demonstrating that the simulation tool can be used to analyse the crash behaviour of subfloor designs on water. This was achieved by simulating the impact on water of a structure representative of an aircraft subfloor. The effect of material anisotropy, skin thickness and skin failure on the structural response was demonstrated.

A first step in extending the coupled FE-SPH modelling beyond fluid-structure interaction problems has been the development and validation of an explicit time integration 1D Lagrangian kernel SPH code which in combination with an algorithm to track crack propagation would make the simulation of dynamic brittle fracture problems possible.



## **Acknowledgements**

First of all I would like to thank Rade for giving me the opportunity to undertake this project, and for his support as a supervisor and friend during the project.

Secondly, I want to thank my parents and brother for the nice breaks, moral support, encouragement and care.

Thank you to my friends for the distraction.

Finally thank you Belen, you have helped me get through the last two years.

## Notation

$\nabla$	= gradient operator
$\langle \dots \rangle$	= SPH approximation
$\delta$	= Kronecker delta
$\Delta t$	= time step
$\varepsilon$	= strain
$\dot{\varepsilon}$	= strain rate
$\phi$	= angular displacement
$\nu$	= Poisson's ratio
$\rho$	= density
$\sigma$	= Cauchy stress
$\sigma_Y$	= yield stress
$a$	= acceleration
$A$	= area, anisotropy coefficient matrix
$b$	= body force
$B$	= strain-displacement/material-compatibility matrix
$c$	= Speed of sound
$C$	= material compliance
$e$	= Internal energy
$E$	= Young's modulus/Specific internal energy
$E_k$	= kinetic energy
$E_p$	= potential energy
$f$	= force vector
$G$	= shear modulus
$h$	= SPH smoothing length
$J$	= Jacobian matrix
$K$	= stiffness matrix, contact stiffness parameter
$m$	= nodal/particle mass
$M$	= mass matrix
$N$	= shape function/unit normal vector
$NP$	= number of particles
$s$	= deviatoric stress tensor
$t$	= time
$u$	= translational displacement
$v$	= translational velocity vector
$V$	= volume
$W$	= SPH kernel function/work
$x$	= position, current position

# CONTENTS

<b>CHAPTER 1 INTRODUCTION.....</b>	<b>1</b>
1.1 STRUCTURE OF THE THESIS .....	1
1.2 WATER IMPACT .....	2
1.3 HYDROCODE MODELLING .....	4
<b>CHAPTER 2 GALERKIN DERIVATION OF THE SPH EQUATIONS.....</b>	<b>6</b>
2.1 LITERATURE SURVEY .....	6
2.1.1 <i>Smoothed Particle Hydrodynamics</i> .....	7
2.1.2 <i>Tensile Instability</i> .....	8
2.1.3 <i>Lack of Consistency</i> .....	9
2.1.4 <i>Zero-energy modes</i> .....	11
2.1.5 <i>Boundary Conditions</i> .....	12
2.2 GALERKIN DERIVATION OF DISCRETISED MOMENTUM EQUATIONS .....	15
2.3 THE GENERAL FORM OF THE [N] AND [B] MATRICES IN 3-D .....	19
2.4 THE GENERAL FORM OF [M] <sub>ij</sub> AND [K] <sub>ij</sub> IN A 1-D SPH CASE .....	20
2.5 DERIVATION OF NORMALISATION AND CORRECTED GRADIENT CONDITIONS .....	24
2.5.1 <i>Introduction</i> .....	24
2.5.2 <i>Homogeneity of Space</i> .....	25
2.5.3 <i>Isotropy of Space</i> .....	26
2.6 COMPARISON OF ACCELERATION CALCULATION WITH ‘CLASSICAL’ SPH FORMULATION.....	28
2.7 EXPLICIT FORMULATION OF THE SPH EQUATIONS .....	29
2.7.1 <i>The Conservation of Momentum Equation</i> .....	32
2.7.2 <i>Strain Rate</i> .....	37
<b>CHAPTER 3 CONTACT MODELLING IN MESHLESS METHODS .....</b>	<b>39</b>
3.1 INTRODUCTION .....	39
3.1.1 <i>Kuhn-Tucker Conditions</i> .....	39
3.1.2 <i>General Variational Framework and Conservation of Energy and Momentum</i> .....	41
3.1.3 <i>Contact in Meshless Methods</i> .....	45
3.2 NEW CONTACT ALGORITHM.....	46
3.2.1 <i>Monaghan’s Repulsive Force</i> .....	46
3.2.2 <i>Contact Potential and Contact Force</i> .....	47
3.2.3 <i>Variational Consistency of the Proposed Contact Algorithm</i> .....	50
3.3 NUMERICAL EXAMPLES .....	51
3.3.1 <i>1D Wave Propagation</i> .....	51
3.3.2 <i>1D Bar Impact</i> .....	53
3.3.3 <i>2D Plate Impact</i> .....	58
3.3.4 <i>Fluid Sloshing in 2D</i> .....	59
3.3.5 <i>3D Hypervelocity Impact</i> .....	61
3.4 CONCLUSIONS .....	62
<b>CHAPTER 4 COUPLING OF SPH SOLVER WITH DYNA3D .....</b>	<b>64</b>
4.1 INTRODUCTION .....	64
4.1.1 <i>Previous work on coupling of SPH and FE</i> .....	64
4.2 STRUCTURE OF DYNA3D .....	65
4.2.1 <i>Basic Structure of DYNA3D</i> .....	66
4.2.2 <i>Data handling</i> .....	67
4.2.3 <i>Solution Section</i> .....	68
4.3 STRUCTURE OF SPH SOLVER .....	70
4.3.1 <i>Basic Structure of the SPH solver</i> .....	70
4.3.2 <i>Data Handling</i> .....	71
4.3.3 <i>Solution Section</i> .....	72
4.4 LINKING OF SPH AND DYNA3D CODES.....	73

4.4.1 Basic Structure of combined DYNA3D – SPH code.....	73
4.4.2 Data Handling .....	74
4.4.3 Solution Section .....	75
4.4.4 Post Processing.....	75
4.4.5 Contact Potential and Contact Force .....	77
4.5 VALIDATION TESTS .....	79
4.5.1 Wave propagation test .....	79
4.5.2 Plate Impact.....	81
<b>CHAPTER 5 ANISOTROPIC PLASTICITY .....</b>	<b>84</b>
5.1 INTRODUCTION .....	84
5.2 THEORY .....	85
5.2.1 Homogeneous Functions.....	85
5.2.2 Yield Criterion .....	85
5.2.3 Flow Rule.....	88
5.2.4 Hardening .....	89
5.2.5 Determination of the parameter $\hat{\lambda}$ .....	92
5.2.6 Strain-Rate dependency .....	93
5.2.7 Equation of State.....	94
5.2.8 Spall .....	95
5.3 MATERIAL TESTING.....	95
5.4 IMPLEMENTATION IN LLNL-DYNA3D .....	98
5.4.1 Overview of LLNL-DYNA3D F3DM33 routine .....	98
5.4.2 Implementation of the New Material Model .....	99
5.4.3 Validation of Initial Anisotropic Model .....	101
5.4.4 Implementation of Kinematic Hardening.....	102
5.4.5 Validation of Combined Isotropic-Kinematic Hardening Model.....	103
5.4.6 Implementation of Strain Rate Dependency.....	104
5.4.7 Validation of Strain Rate Dependency.....	107
5.4.8 Implementation of Equation of State.....	109
5.4.9 Validation of the Implementation of the Equation of State .....	110
5.4.10 Implementation of Spall Models.....	112
5.4.11 Validation of the spall models.....	113
5.5 SIMULATION OF PLATE IMPACT ON THE ALUMINIUM ALLOY 7010 – T6 .....	113
5.5.1 Introduction .....	113
5.5.2 Experimental Setup .....	114
5.5.3 DYNA3D Model .....	115
5.5.4 Simulation Results.....	117
5.5.5 Conclusion .....	122
<b>CHAPTER 6 WATER IMPACT OF RIGID CYLINDRICAL SPECIMEN.....</b>	<b>123</b>
6.1 INTRODUCTION .....	123
6.1.1 Test Setup.....	123
6.2 SIMULATION RESULTS .....	125
6.2.1 Model Description and Sensitivity Study .....	125
6.2.2 Influence of EOS.....	129
6.2.3 Influence of Contact Stiffness.....	131
6.2.4 Influence of the discretisation density.....	133
6.2.5 Influence of the volume of water discretised.....	134
6.2.6 Influence of Smoothing Length .....	134
6.2.7 Influence of Young's Modulus of Wood .....	134
6.2.8 Comparison of Experimental and Simulation Results.....	136
<b>CHAPTER 7 APPLICATIONS – WATER IMPACT OF METALLIC STRUCTURES.....</b>	<b>142</b>
7.1 SUBFLOOR DESCRIPTION .....	142
7.2 MODELLING APPROACH .....	142
7.3 SIMULATION RESULTS – RIGID SURFACE.....	144

7.3.1 Isotropic Material Model.....	144
7.3.2 Anisotropic Material Model.....	148
7.4 SIMULATION RESULTS – WATER IMPACT .....	152
7.4.1 isotropic .....	152
7.4.2 Anisotropic.....	156
7.4.3 Skin Failure.....	160
<b>CHAPTER 8 1D LAGRANGIAN SPH .....</b>	<b>166</b>
8.1 LAGRANGIAN KERNEL.....	166
8.2 1D LAGRANGIAN SPH CODE .....	168
8.3 SIMULATION RESULTS .....	172
8.3.1 Wave Propagation in Elastic Bar .....	172
8.3.2 Symmetric Bar Impact.....	173
8.3.3 Plate Impact Simulation.....	176
<b>CHAPTER 9 CONCLUSION.....</b>	<b>181</b>
9.1 CONCLUSION .....	181
9.2 RECOMMENDATIONS FOR FURTHER WORK.....	181
<b>REFERENCES .....</b>	<b>183</b>
<b>APPENDIX A. CLASSICAL SPH DERIVATION .....</b>	<b>189</b>
<b>APPENDIX B. IMPLEMENTATION OF DYNA3D-SPH COUPLING .....</b>	<b>192</b>
DESCRIPTION OF THE ENSIGHT CASE DATA FORMAT.....	192
DESCRIPTION OF THE DYNA3D DATA STRUCTURE.....	197
DESCRIPTION OF THE MCM DATA STRUCTURE.....	203
DESCRIPTION OF THE COMBINED DYNA3D-SPH OUTPUT SECTION .....	203
COUPLING ALGORITHM .....	206
<b>APPENDIX C. DETAILED DESCRIPTION OF F3DM33.....</b>	<b>209</b>

## LIST OF FIGURES

Figure 1.1: Load Path during Ground Impact.....	3
Figure 1.2: Load Path during Water Impact.....	4
Figure 2.1: Zero Energy Modes.....	12
Figure 2.2: Central Difference Time Integration Loop. ....	31
Figure 3.1: Schematic representation of two bodies in contact.....	40
Figure 3.2: Reference configuration advocated by Simo and Tarnow [ 81 ]. ....	44
Figure 3.3 Schematic representation of the penetration. ....	47
Figure 3.4: Velocity Profile for Wave Propagation Problem, at time = 128.8 $\mu$ s.....	52
Figure 3.5: Velocity Profile for Block Wave Propagation Problem, at time = 128.8 $\mu$ s. .....	53
Figure 3.6: Schematic representation of the 1D test problem .....	54
Figure 3.7: Pressure profiles with and without contact at 48 $\mu$ s. ....	54
Figure 3.8: Velocity profiles with and without contact at 48 $\mu$ s. ....	55
Figure 3.9: Pressure profiles with and without contact at 300.0 $\mu$ s. ....	55
Figure 3.10: Velocity profiles with and without contact at 300.0 $\mu$ s. ....	56
Figure 3.11: Kinetic - and Internal Energy Time Histories without contact. ....	56
Figure 3.12: Kinetic - and Internal Energy Time Histories with contact. ....	57
Figure 3.13: Symmetric Block Impact in 2D: Problem Setup.....	58
Figure 3.14: Pressure plots 1 $\mu$ s after impact, without contact (LHS) and with contact (RHS).....	59
Figure 3.15: Pressure plots 2 $\mu$ s after impact, without contact (LHS) and with contact (RHS).....	59
Figure 3.16: Materials plots 5 $\mu$ s after impact, without contact (LHS) and with contact (RHS).....	59
Figure 3.17: Fluid Sloshing Simulation: Problem Setup.....	60
Figure 3.18: Position of the fluid at 0.00, 0.21, 0.51 and 1.98 s. ....	60
Figure 3.19: Isometric View of 3D Hypervelocity Impact at 0.0 and 0.2 $\mu$ s. ....	61
Figure 3.20: Cross Section Plots of 3D Hypervelocity Impact at 0.0, 0.1 and 0.2 $\mu$ s.....	62
Figure 3.21: Cross Section Plot of 3D Hypervelocity Impact at 1.5 $\mu$ s. ....	63
Figure 4.1: Basic Structure of DYNA3D. ....	67
Figure 4.2: Structure of main solution subroutine.....	69
Figure 4.3: Main steps in the calculation of the force vector of hexahedron elements, for strength model requiring an equation of state.....	70
Figure 4.4: Basic Structure of SPH solver.....	71
Figure 4.5: Structure of main solution loop of SPH solver. ....	72
Figure 4.6: Main steps in the calculation of the acceleration vector for SPH particles, for a strength model requiring an equation of state. ....	73
Figure 4.7: Basic Structure of Coupled DYNA3D and SPH code. ....	74
Figure 4.8: Structure of main DYNA3D – SPH solution subroutine. ....	76
Figure 4.9: FE and SPH (without contact) Velocity Profiles of 3D Wave Propagation Simulation, at 128.2 $\mu$ s.....	80
Figure 4.10: FE and SPH (with contact) Velocity Profiles of 3D Wave Propagation Simulation, at 128.2 $\mu$ s.....	80

Figure 4.11: Coupled FE-SPH and SPH-FE Velocity Profiles of 3D Wave Propagation Simulation, at 128.2 $\mu$ s.....	81
Figure 4.12: Plate Impact Velocity Profiles at 48 $\mu$ s.....	82
Figure 4.13: Plate Impact Velocity Profiles at 48 $\mu$ s.....	83
Figure 5.1: Punch Stretching Arrangement [ 75 ]. ....	97
Figure 5.2: Different Types of Marking of the Bulge Test Specimens [ 75 ]. ....	98
Figure 5.3: Photos of the Test Specimens Marking before and after the test [ 75 ]. ....	98
Figure 5.4: Anisotropic vs Model 3 Comparison: Isotropic Hardening.....	102
Figure 5.5: Anisotropic vs Model 3 Comparison: Kinematic Hardening. ....	104
Figure 5.6: Anisotropic vs Model 19 Comparison: Isotropic Hardening.....	108
Figure 5.7: Anisotropic vs Model 19 Comparison: Kinematic Hardening. ....	109
Figure 5.8: Mesh used for EOS Validation. ....	111
Figure 5.9: Pressure Time history during Plate Impact.....	111
Figure 5.10: Validation of the EOS Implementation in Material Model 53.....	112
Figure 5.11: Validation of the Spall Model Implementation in Material Model 53. ..	113
Figure 5.12: Test Configuration. ....	115
Figure 5.13: FE Mesh. ....	117
Figure 5.14: 234 m/s Impact – Rolling Direction.....	118
Figure 5.15: 234 m/s Impact – Transverse Direction.....	119
Figure 5.16: 450 m/s Impact – Rolling Direction.....	119
Figure 5.17: 450 m/s Impact – Transverse Direction.....	120
Figure 5.18: 895 m/s Impact – Rolling Direction.....	120
Figure 5.19: 895 m/s Impact – Transverse Direction.....	121
Figure 6.1: Rigid Cylinder used in Water Impact Test, from [ 1 ]. ....	123
Figure 6.2: Technical Drawing of Rigid Cylinder used in Water Impact Test, from [ 1 ]. .....	124
Figure 6.3: Water Impact of Rigid Cylindrical Specimen, Experiment, [ 1 ]).....	125
Figure 6.4: Water Impact of Rigid Cylindrical Specimen, Side view of Model (t = 0ms). .....	126
Figure 6.5: Water Impact of Rigid Cylindrical Specimen, Side view of Model (t = 10ms, 20ms, 30ms, 40ms).....	127
Figure 6.6: Water Impact of Rigid Cylindrical Specimen, Isometric of Model (t = 0ms). .....	127
Figure 6.7: Water Impact of Rigid Cylindrical Specimen, Isometric of Model (t = 10ms, 20ms, 30ms, 40ms).....	128
Figure 6.8: Water Impact of Rigid Cylindrical Specimen, location of time history nodes. .....	128
Figure 6.9: Initial model results, acceleration time histories.....	129
Figure 6.10: Murnaghan EOS results (B = 0.000016).....	130
Figure 6.11: Murnaghan EOS results (B = 0.000060).....	130
Figure 6.12: Murnaghan EOS results (B = 0.000180).....	131
Figure 6.13: Increased Contact Stiffness Results.....	133
Figure 6.14: Reduced Contact Stiffness Results.....	133
Figure 6.15: Finer Mesh Results.....	134
Figure 6.16: Increased Volume of Water Results.....	135
Figure 6.17: Smoothing length equal to particle spacing results.....	135
Figure 6.18: Properties of wood.....	136

Figure 6.19: Comparison of Experimental and FE and SPH Simulation Results of Cylinder Impact at 10.6m/s .....	137
Figure 6.20: Comparison of FE and SPH Velocity Time Histories of Cylinder Impact at 10.6m/s .....	138
Figure 6.21: Comparison of Experimental and Simulation Results of Cylinder Impact at 3.3m/s .....	139
Figure 6.22: Comparison of Experimental and Simulation Results of Cylinder Impact at 7.8m/s .....	140
Figure 6.23: Comparison of Experimental and Simulation Results of Cylinder Impact at 10.6m/s .....	141
Figure 7.1: Typical Helicopter Subfloor Box .....	143
Figure 7.2: Subfloor FE Model .....	143
Figure 7.3: Positions where Time histories are recorded. ....	145
Figure 7.4: Time histories for Subfloor Impact, Simulation 1. ....	146
Figure 7.5: Influence of Filtering on Time histories for Subfloor, Simulation 1. ....	146
Figure 7.6: Velocity Time histories for Subfloor, Simulation 1.....	147
Figure 7.7: Effective Plastic Strain [0-10%] in subfloor at 40 ms, Simulation 1.....	147
Figure 7.8: Effective Plastic Strain [0-0.1%] in Longerons at 40 ms, Simulation 1.....	148
Figure 7.9: Time histories for Subfloor, Simulation 2. ....	150
Figure 7.10: Influence of Filtering on Time histories for Subfloor, Simulation 2. ....	150
Figure 7.11: Comparison of CFC600 filtered Time histories for Simulation 1 & 2. ..	151
Figure 7.12: Effective Plastic Strain [0-10%] in Subfloor at 40 ms, Simulation 2. ....	151
Figure 7.13: Effective Plastic Strain [0-10%] in Longerons at 40 ms, Simulation 2.....	152
Figure 7.14: Initial Model Configuration, Simulation 3.....	153
Figure 7.15: Time histories for Subfloor, Simulation 3. ....	153
Figure 7.16: Influence of Filtering on Time histories for Subfloor, Simulation 3. ....	154
Figure 7.17: Deformed Shape after 40ms, Simulation 3. ....	155
Figure 7.18: Effective Plastic Strain (0-10%) after 40ms, Simulation 3.....	155
Figure 7.19: Mode of Collapse of Longerons, Simulation 3. ....	156
Figure 7.20: Lower Skin Deformation after 40ms, Simulation 3.....	156
Figure 7.21: Time histories for Subfloor, Simulation 4. ....	157
Figure 7.22: Influence of Filtering on Time histories for Subfloor, Simulation 4. ....	157
Figure 7.23: Comparison between Time histories of Simulation 3 & 4.....	158
Figure 7.24: Deformed Shape at 40ms, Simulation 4. ....	159
Figure 7.25: Effective Plastic Strain (0-10%) at 40ms, Simulation 4. ....	159
Figure 7.26: Mode of Collapse of Longerons, Simulation 4. ....	160
Figure 7.27: Lower Skin Deformation at 40ms, Simulation 4. ....	160
Figure 7.28: Overview of Time histories, Simulations 3, 5a, 5b.....	161
Figure 7.29: Failed Elements after 12 ms, Simulation 5b. ....	162
Figure 7.30: Cross Section after 40 ms, Simulation 5.....	162
Figure 7.31: Isometric View after 40 ms, Simulation 5. ....	163
Figure 7.32: Side View of Subfloor after 40 ms, Simulation 5.....	163
Figure 7.33: Longerons Deformation after 40 ms, Simulation 5.....	164
Figure 7.34: Effective Plastic Strain [0 – 10%] after 40 ms, Simulation 5. ....	164
Figure 7.35: Skin Failure after 60 ms, Simulation 5. ....	165
Figure 8.1 Central Difference Time Integration Loop for Lagrangian Kernel SPH. ..	170
Figure 8.2 Velocity Profile for Wave Propagation Problem with Lagrangian Kernel, at time = 128.8μs.....	173



Figure 8.3: Schematic representation of the 1D test problem .....	173
Figure 8.4: SPH velocity profiles with Lagrangian and Eulerian SPH at 48.0 $\mu$ s. ....	174
Figure 8.5: SPH pressure profiles with Lagrangian and Eulerian SPH at 48.0 $\mu$ s. ....	174
Figure 8.6: SPH velocity profiles with Lagrangian and Eulerian SPH at 128.8 $\mu$ s .....	175
Figure 8.7: SPH pressure profiles with Lagrangian and Eulerian SPH at 128.8 $\mu$ s .....	175
Figure 8.8: Kinetic - and Internal Energy Time Histories with Lagrangian SPH. ....	176
Figure 8.9: Lagrangian Kernel SPH Plate Impact Simulations - 234ms. ....	178
Figure 8.10: Lagrangian Kernel SPH Plate Impact Simulations - 450ms. ....	179
Figure 8.11: Lagrangian Kernel SPH Plate Impact Simulations - 895ms. ....	179
Figure B.1: Example of .case file. ....	192
Figure B.2: Example of .geo file. ....	195
Figure B.3: Example of nodal variable data file.....	196
Figure B.4: Example of element variable data file.....	197
Figure B.5: Types of Solid Elements. ....	199
Figure B.6: Types of Shell Elements.....	200
Figure B.7: Pointers from input element number to position stress data in auxvec. ....	201
Figure B.8: DYNA3D Data Structure. ....	202

LIST OF TABLES

Table 3.1: Elastic Material Constants for Wave Propagation Test..... 51

Table 5.1: Overview of modified DYNA3D subroutines..... 100

Table 5.2: Material Model 3 Constants for Steel..... 101

Table 5.3: Overview of DYNA3D Routines Modified for Strain Rate Dependency . 107

Table 5.4: Material Parameters for Steel ..... 108

Table 5.5: Materials Data for Al 7010-T6 and Al 6082-T6. .... 114

Table 5.6: Quasi-static mechanical properties of 7010-T6..... 116

Table 5.7: Material Properties used in Analysis..... 116

Table 6.1: Measured Acceleration during Water Impact Experiments. .... 125

Table 6.2: Grüneisen EOS parameters for water..... 126

Table 6.3: Comparison of Simulated Acceleration Peaks to Experimental Data..... 141

Table 7.1: Dimension of Subfloor Model Components ..... 143

Table 7.2: Plate Thickness of Subfloor Model Components..... 144

Table 7.3: Summary of Subfloor Simulations Setup..... 144

Table 7.4: Material Constants for Al 2024, Simulations 1 & 3..... 144

Table 7.5: Acceleration Levels for Subfloor, Simulation 1..... 145

Table 7.6: Anisotropic Material Properties for Al 2024..... 149

Table 7.7: Acceleration Levels for Subfloor, Simulation 2..... 149

Table 7.8: Acceleration Levels for Subfloor, Simulation 3..... 154

Table 7.9: Acceleration Levels for Subfloor, Simulation 4..... 158

Table 8.1: Summary of Results ..... 180

Table B.1: Overview of Pointers to Nodal Field Variables and Element Connectivity  
Data in the Main Database..... 198

Table B.2: Storage of Nodal Positions in the Main Database. .... 198

Table B.3: Storage of Hexahedral Element Connectivity in the Main Database ..... 199

Table B.4: Storage of Pentahedral Element Connectivity in the Main Database..... 199

Table B.5: Storage of Tetrahedral Element Connectivity in the Main Database. .... 199

Table B.6: Storage of Quadrilateral Shell Element Connectivity in the Main Database.  
..... 200

Table B.7: Storage of Triangular Shell Element Connectivity in the Main Database. 200

Table B.8: Storage of Beam Element Connectivity in the Main Database. .... 200

Table B.9: Pointer to Element Activity Flag ..... 203

Table B.10: MCM data format for Particle Variables..... 203

## Chapter 1 Introduction

The purpose of this thesis is to develop a software tool that can be used to analyse water impact. Two facets that are of importance when numerically modelling the impact of metallic structures on water are metal anisotropy and water behaviour during impact. In order to study these effects, an appropriate anisotropic material model needs to be available, and a simulation technique capable of dealing with the complex interaction between water and a collapsing structure has to be used.

The simulation codes used throughout this thesis are the Cranfield version of the Lawrence Livermore National Laboratories DYNA3D finite element code, and the Cranfield University's Structures and Materials group smooth particle hydrodynamics (SPH) code MCM.

The material presented in the chapters on the contact modelling, FE-SPH coupling and material modelling discusses the implementation in high detail. The reason for this is that these chapters serve a dual purpose. First of all, they present the work that was required for this research, but secondly these chapter are also intended as an aid to making changes to the public domain version of DYNA3D.

### **1.1 Structure of the Thesis**

Chapter 2 presents an overview of meshless methods, in particular the SPH method. The SPH equations are derived in a Galerkin framework. The finite difference time integration scheme for the semidiscretised equations is provided.

In Chapter 3 an overview of contact modelling techniques is given. A method of modelling contact based on a contact potential is proposed. The method does not require surfaces to be constructed, and avoids problems with non-uniqueness of the contact normal at corners. It is demonstrated how the method can be incorporated into a Galerkin derivation of the SPH equations. The performance of the method is illustrated with various examples in 1, 2 and 3D.

Chapter 4 deals with the coupling of an SPH solver with DYNA3D. A novel approach has been used which treats FE nodes as particles in the contact section. The contact algorithm described in Chapter 3 is used to deal with the interaction between SPH particles and FE nodes. The coupling algorithm is validated by modelling wave propagation and plate impact problems.

Chapter 5 provides an overview of the background theory required for an anisotropic plasticity model. Apart from the anisotropic yield criterion the model also incorporates combined isotropic-kinematic hardening, strain rate effects and an equation of state. Strain rate effects can influence the material response during dynamic loading, in particular during impact events. In order to validate the implementation of the model into DYNA3D plate impact experiment data was used. This required the implementation of an equation of state to deal with the material behaviour at high

pressures. Simple failure models to model the spall failure observed in the experiments where also included.

In Chapter 6 the comparison of coupled FE-SPH simulation results to experimental data is presented for the case of the impact of a rigid half-cylinder on water. The experimental data consists of acceleration time histories at two points on the top of the impactor. Three sets of acceleration time histories of impacts at three different speeds (3.3, 7.8 and 10.6 m/s) are used. A series of model parameters is varied to determine a good model setup. The simulation results show good agreement with the experimental data as far as the acceleration pulse upon impact is concerned. The length of this pulse tends to be underestimated.

The validation of the coupled FE-SPH code is continued in Chapter 7, and combined with the use of the anisotropic plasticity model implemented in Chapter 5. A series of simulations of a structure representative of a helicopter subfloor is performed. The influence of metal anisotropy on the acceleration time histories recorded on the loading points of the subfloor is demonstrated for an impact on a rigid surface. These simulations also serve as a reference for the equivalent water impact simulations. It is shown that in this case the peak acceleration is not affected by metal anisotropy. To complete this validation two simulations are performed with a lower skin with reduced thickness. In one of these simulations a set of elements is given a low failure strain to allow the skin to tear and water to enter the structure. A full validation against experimental results was not possible as no data was available.

Chapter 8 presents the background theory, development and implementation of a 1D Lagrangian kernel SPH code. By using a Lagrangian kernel the tensile instability present in SPH can be removed. The use of a Lagrangian kernel would allow for example for the modelling of brittle fracture in solids, or the improved modelling of wave propagation in solids and fluids. The implementation in a 3D code with explicit time integration is complex due to the need to calculate material rotations accurately. As there are no material rotations in 1D this problem is removed. The plate impact problem used to validate the anisotropic plasticity model in Chapter 5 along with simpler validation problems, has been used to validate the 1D SPH code. The correct resolution of the stress time histories, the spall signals in particular, require tensile waves to be modelled accurately. The simulation results are in good agreement with the experiment.

## **1.2 Water Impact**

Aircraft crash or ditching frequently occurs on water or soft soil. For example 25% of helicopter accidents in the UK happened on water [ 86 ]. The mechanisms involved in a water impact can result in a more severe crash situation than an impact on solid surface for the same velocity [ 86 ]. The following aspects contribute to this [ 86 ]:

- ▶ The ineffectiveness of the undercarriage to absorb the impact energy.
- ▶ The poor hydrodynamic performance due to the fuselage shape.
- ▶ The response of the structure is dictated by water pressures rather than inertia forces.

At this moment little or no attention has been paid to improving the water crashworthiness performance of aircraft. Improvements in the subfloor design could improve the crash behaviour on water, and hence improve the survivability of a crash event. In order to achieve this two problems need to be addressed. The first being the reduction of the acceleration levels the occupants are subjected to in order to minimise the risk of incapacitating injuries. The second is the equipment of aircraft with flotation devices in order to avoid casualties through drowning.

When impacting on solid surfaces the loads distribute through the structural elements of the under carriage and subfloor frames and longerons (see Figure 1.1). These elements crush progressively. When impacting a water surface, the water pressure on the aircraft skin panels are loaded (see Figure 1.2). This can cause the skin to fail. The skin is designed to carry aerodynamic loads, but all structural loads are transmitted through frames and longerons. The skin failure can have two negative effects. Firstly it can break the floor panels allowing water to enter the fuselage. This results in a reduction of flotation time [ 86 ]. Secondly the load path will change completely as the subfloor frames will not absorb energy. As a consequence the loads on the passengers and upper structures may increase. The deformation of the upper structure may also cause the doors to either fall off or jam. In the first case this would reduce the flotation capability of the aircraft severely, while in the second case it would jeopardise the evacuation of the cabin. In order to reduce the severity of water impacts it is necessary to achieve a controlled structural collapse of the floor. This would require preventing the rupture of the outer skin. If the outer skin is load bearing and does not fail the loads can be distributed more efficiently and in a controlled manner into the structure.

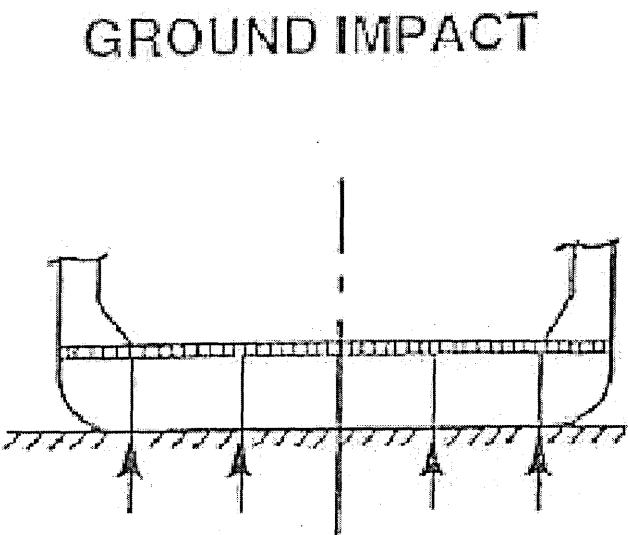


Figure 1.1: Load Path during Ground Impact.

# WATER IMPACT

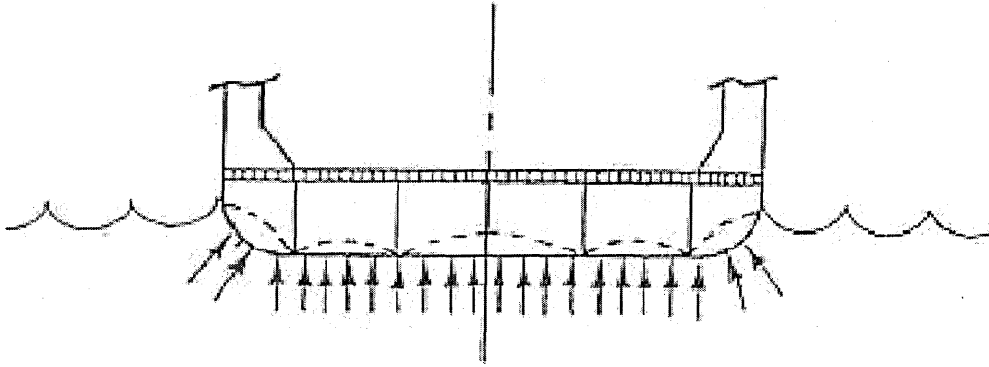


Figure 1.2: Load Path during Water Impact

## 1.3 Hydrocode Modelling

In order to design an aircraft such that it is crashworthy on water a series of tests, simulations or a combination of both would be required. The destructive tests required to evaluate the level of performance achieved by a specific design are particularly expensive because of the destructive nature of the test. Therefore there is a need for reliable simulation tools.

A hydrodynamics simulation code, or hydrocode, is a simulation code for solving large deformation transient problems that occur on a short time scale. Hydrocodes are based on the conservation of mass, momentum and energy equations. These equations can be solved either using a Eulerian, Lagrangian or hybrid spatial discretisation.

The advantage of an Eulerian mesh is that large deformations can be handled easily. This makes it suitable to model fluid flow. The disadvantage is that it needs an algorithm to track material interfaces. It also requires an accurate advection algorithm to prevent material loss through diffusion, and an algorithm to track material history variables. These algorithms make Eulerian codes computationally expensive. For these reasons it is not suited to model structures.

A Lagrangian mesh has as advantages that material boundaries are accurately defined, material history variables are easily tracked and structural elements (shell and beam elements) can be used. The disadvantage of a Lagrange mesh is that large deformations can cause problems through excessive mesh distortion and that contact algorithms are required to deal with the interaction of different bodies.

The hybrid approaches are the Arbitrary Lagrangian Eulerian (ALE) method or Euler-Lagrange coupling. In the ALE method an element is allowed to deform (Lagrangian), but material is also allowed to flow in and out of the element (Eulerian). This method combines the advantages of both methods, but it also inherits the disadvantages of both methods (mesh distortion and diffusion) even though their effects should be less pronounced. Coupled Euler-Lagrange codes allow some parts to be modelled with an Eulerian grid and others with a Lagrangian mesh.

Meshless methods, for example the SPH method, allow very large deformations in a Lagrangian framework. Therefore they potentially are an interesting alternative for the methods described above. Their disadvantage is that they are computationally expensive and can require a background grid for the integration of the discretised equations. The SPH method is completely meshless as it does not require a background grid. Unfortunately SPH suffers from tensile instabilities and zero energy modes. When used to model a fluid the tensile instability is less important. This makes coupled FE-SPH a potential solution for modelling water impact of deforming and collapsing structures. As the method is Lagrangian no material is lost through diffusion. Apart from the numerical problems associated with the SPH method, the treatment of boundary conditions, Von Neumann, Dirichlet or contact, is also still a topic of research.

In this thesis a simulation tool will be developed which allows for the modelling of water impact by using a combination of a Lagrangian finite element code for the structure and an SPH solver to model the water.

## Chapter 2 Galerkin Derivation of the SPH Equations

### 2.1 Literature Survey

The finite element method has traditionally been the most popular and widely used method in engineering computations. The continuum is discretised in elements that are connected by a topological map, a mesh. The field variables are interpolated upon the mesh using the finite element interpolation functions, or shape functions. This ensures the compatibility of the mesh. This is the strength and at the same time, in certain cases, the weakness of the finite element method. For example, in Lagrangian computations the mesh may get very distorted. This can result in the error termination of the analysis, or a drastic deterioration of the results.

One solution is the use of adaptive remeshing procedures [ 17 ]. For example in impact or penetration problems and fluid structure interaction. However this procedure is very complex, and frequently very time consuming. Furthermore the difficulty lies not only in the remeshing procedure itself, but also in the mapping of state variables from the old to the new mesh. This can lead to significant errors because of the diffusive nature of this process. Hence a frequent remesh is not desirable. A possible solution is the use of Arbitrary Lagrangian Eulerian (ALE) formulations [ 62 ], [ 7 ]. In this case the mesh is not moved with the material, a certain amount flow of material through the elements is allowed of the material in an attempt to minimise the mesh distortion. However due to the partially Eulerian nature severe errors can still be introduced. For example in the case of very large strain or high speed structural behaviour.

An other problem is that the mesh may carry an inherent bias in certain computations. This bias can lead to errors in the calculations. Examples are the strain localisation problem [ 73 ], [ 74 ], and crack propagation.

Therefore it would be advantageous for these types of problems if the continuum could be discretised as a set of nodal points, without the constraints of a mesh. This is the main reason for the current research into meshless methods. A second reason is the possible reduction in effort to discretise the continuum. The finite element mesh generation for complex structures or mechanisms can be very time consuming, and can take times longer than the actual numerical solution of the model.

The advantages of meshfree methods can be summarised as follows:

- ▶ They can easily handle very large deformations because the nodal connectivity can change in time.
- ▶ It is easier to use the method starting from a CAD geometry model than FE, because there is no need to generate a mesh.
- ▶ It is easier to handle damage, failure and fracture of components. There are no preferential directions, crack propagation is easier.
- ▶ The accuracy can be controlled more easily, nodes can be added (h-adaptivity) where more detailed solutions are required.



- The method can incorporate fine scale solutions into a coarse scale (Multi Resolution analysis) [ 63 ], [ 64 ], [ 65 ].

Several particle methods have been proposed in the literature. Some of them start from the strong form of a PDE. For example the Smoothed Particle Hydrodynamics (SPH) method. Recently Libersky and Randles [ 60 ], [ 76 ] have developed a spatially staggered meshfree method called Dual Particle Dynamics (DPD). This method is also derived from the strong form, and uses MLS interpolations in combination with a spatial staggering of stress and velocity points. They also use a predictor-corrector time integration scheme. Other methods start from the weak form of the PDE. For example the Element Free Galerkin Method (EFGM) [ 6 ], the Reproducing Kernel Particle Method (RKPM) [ 66 ], [ 67 ], the h-p Cloud Method [ 68 ], the meshless local Petrov-Galerkin Method (MLPG) [ 2 ], [ 3 ] fall under this category.

### 2.1.1 Smoothed Particle Hydrodynamics

Smoothed Particle Hydrodynamics is one of the earliest particle methods in computational mechanics. The Smoothed Particle Hydrodynamics Method (SPH) was developed simultaneously by Lucy [ 69 ], and Gingold [ 36 ] and Monaghan in 1977 [ 70 ]. Their interest was in the simulation of astrophysical problems. For example the formation of galaxies or the collision of galaxies. The movement of those particles is similar to the movement of a gas or liquid and can be modelled using the Newtonian hydrodynamics equations. Because of its distinct advantages the SPH method was quickly adopted to model problems of applied mechanics in general not just hydrodynamical problems. The SPH method was extended to work with the full stress tensor by Libersky, Randles, Petschek, Carney et al. [ 58 ], [ 59 ], [ 77 ]. Their main field of interest was the simulation of hypervelocity impact and detonation and penetration problems. Their results demonstrated the advantages of the SPH method in this type of applications over Lagrangian or Eulerian Finite Element codes. Johnson and Stryk [ 46 ], [ 82 ] also applied the SPH method to penetration problems. Other examples of where the SPH method has been used are compressible flow [ 85 ], multiphase flow [ 38 ], [ 22 ], heat conduction [ 27 ] and hydrodynamic ram [ 18 ].

The SPH method is a popular tool for the simulation of these types of problems despite its shortcomings:

- tensile instability
- lack of consistency
- zero energy mode
- treatment of boundary conditions

The shortcomings will be discussed briefly after the basic concept and equations of the SPH method have been described.

The SPH method is based on the convolution principle or interpolant integral. Thus, any function  $f(x')$  can be evaluated at  $x$  by its smoothed value  $\langle f(x) \rangle$  given by

$$\langle f(x) \rangle = \int_{\Omega} f(x') W(x - x', h) dx' \quad (\text{Eq. 2.1})$$

Where  $W(x - x', h) = W(r, h)$  is the kernel function,  $\Omega$  is the compact support of the kernel and  $h$ , also called the smoothing length, defines size of the support. In computations one employs kernels with compact support such as splines. The kernel  $W(r, h)$  should possess the following properties:

$$W(x - x', h) \geq 0 \quad (\text{Eq. 2.2})$$

$$\int_{\Omega} W(x - x', h) dx' = 1 \quad (\text{Eq. 2.3})$$

$$W(x - x', h) \xrightarrow{h \rightarrow 0} \delta(x - x') \quad (\text{Eq. 2.4})$$

$$W(x - x', h) \in C^p(\mathbb{R}^n), p \geq 1 \quad (\text{Eq. 2.5})$$

Where  $\delta(x - x')$  is the Kronecker delta function. This third property ensures convergence, while the last property comes from the requirement that the kernel has to be differentiable to a certain order depending on the PDE that is being discretised.

In numerical applications the integral in Eq. 2.1 is approximated by point integration:

$$\langle f(x) \rangle \approx \sum_{j=1}^{NP} \frac{m_j}{\rho_j} f(x_j) W(x - x_j, h) \quad (\text{Eq. 2.6})$$

The spatial derivative of  $f$  can be calculated using:

$$\langle \nabla f(x) \rangle \approx \sum_{j=1}^{NP} \frac{m_j}{\rho_j} f(x_j) \nabla W(x - x_j, h) \quad (\text{Eq. 2.7})$$

### 2.1.2 Tensile Instability

A Von Neumann stability analysis of the SPH method was conducted Swegle et al [ 83 ] and Balsara [ 5 ]. This has revealed that the SPH method suffers from a tensile instability. This instability usually manifests itself in computations as an unphysical clumping together of the particles when subjected to a tensile, spherical, stress. From Swegle's stability analysis it emerged that the criterion for stability was that:

$$W''\sigma > 0. \quad (\text{Eq. 2.8})$$

In order to remedy this problem several solutions have been proposed. Randles and Libersky proposed adding dissipative terms, which is related to conservative smoothing [ 77 ]. Dyka [ 30 ] proposed an original solution by using a non-colocated discretisation of stress and velocity points. At one set of points the stresses are evaluated, while the momentum equation is calculated at another set of points. The 'stress' points are equivalent to the Gauss quadrature points in FE, the other set of points is equivalent to

the element nodes. This approach was extended to two dimensions, in combination with kernel normalisation, by Vignjevic and Campbell [ 87 ]. Other solutions were proposed by Chen [ 25 ] and Monaghan [ 71 ]. The former proposes a corrective SPH method by enforcing higher order consistency, while the latter proposes the addition of an artificial force to stabilise the computation. Recently Randles and Libersky combined MLS interpolation with the stress and velocity point approach. They called this approach the Dual Particle Dynamics method [ 60 ], [ 76 ]. The issue of stability was dealt with in the context of particle methods in general by Belytschko et al [ 8 ], and independently by Randles, Petschek, Libersky and Dyka [ 76 ]. They reach the same conclusions as Swegle.

### 2.1.3 Lack of Consistency

The problem of consistency has been extensively addressed in the literature. The SPH kernel function satisfies the 0<sup>th</sup> order moment condition (Eq. 2.3). Most kernel functions also satisfy higher order moment conditions:

$$\int_{\Omega} (x' - x)^n W(x - x', h) dx' = 0, n \in \mathbb{N}^+ \quad (\text{Eq. 2.9})$$

So in general one can write the moment conditions as:

$$\int_{\Omega'} (x' - x)^n W(x - x', h) dx' = \delta_{0n}, n \in \mathbb{N}^+ \quad (\text{Eq. 2.10})$$

However, even though these conditions may hold in the continuous form they in general no longer hold in a discretised system:

$$\sum_{j=1}^{NP} \frac{m_j}{\rho_j} (x_j - x_i)^n W(x_i - x_j, h) \neq \delta_{0n}, n \in \mathbb{N}^+ \quad (\text{Eq. 2.11})$$

In order to restore consistency several methods have been proposed. The most widely used ones are kernel gradient correction, the Reproducing Kernel Particle Method (RKPM) and Moving Least Squares (MLS) interpolation. Kernel gradient correction allows gradients of linear functions to be calculated exactly in the discretised system. The advantage of this method is that is computationally relatively cheap. It was introduced by Johnson [ 47 ] and Randles and Libersky [ 77 ], who published a general formulation for three dimensions. The other two methods RKPM and MLS are more computationally expensive because they require the solution of a set of equations for every neighbourhood.. In the RKPM method the kernel is multiplied by a polynomial function:

$$\tilde{W}(x - x_j, h) = C(x, x - x_j) W(x - x_j, h) = \sum_{k=0}^K b_k(x, h) \left( \frac{x - x_j}{h} \right)^k W(x - x_j, h) \quad (\text{Eq. 2.12})$$

This function  $C$  restores  $K^{\text{th}}$  order consistency. This implies that a polynomial of degree  $k$  can be reproduced exactly.

Further, by careful selection of this kernel function, it is possible to ensure that polynomials up to a given degree  $k$  are exactly reproduced by the above integral. To demonstrate this consider a simple one-dimensional case where the function  $f(x_0)$  inside the integral has been approximated as a Taylor series expansion about  $x$  to give,

$$f(x') = f(x) + (x' - x) \frac{\partial f(x)}{\partial x} + \frac{1}{2} (x' - x)^2 \frac{\partial^2 f(x)}{\partial x^2} + \dots \quad (\text{Eq. 2.13})$$

Substituting this expression into Eq. 2.1 gives

$$\langle f(x) \rangle = f(x) \int W(x - x') dx' + \frac{\partial f(x)}{\partial x} \int (x' - x) W(x - x') dx' + \dots \quad (\text{Eq. 2.14})$$

Hence if the following conditions are satisfied:

$$\int_{\Omega'} (x' - x)^n W(x - x', h) dx' = \delta_{0n}, \quad n \in \mathbb{N}^+, \quad 0 \leq n \leq k$$

the approximation is said to be of order  $k$  and  $\langle f(x) \rangle$  will coincide with  $f(x)$  for polynomials of degree equal or lower than  $k$ .

The Moving Least Squares (MLS) interpolation is described by Belytschko [ 6 ] and Dils [ 31 ], [ 32 ]. The principal property of a MLS is to exactly reproduce an arbitrary set of orthogonal functions everywhere in space when the input data consists of these functions evaluated at a given set of points. These points do not have to lie on a regular grid, they may be randomly distributed. This interpolation method was introduced by Lancaster and Salkauskas [ 54 ].

If the set of orthogonal basis functions are stored in a vector  $p^t(x)$ ,  $u_i$  is the nodal value of the data at point  $x_i$ , and  $W_i(x)$  is a smooth weighting function with compact support then the goal is to find an approximation to the nodal data of the form:

$$u^h(x) = p^t(x) \cdot a(x) \quad (\text{Eq. 2.15})$$

Note that  $a(x)$  is a function of the position as well. This is due to the 'moving' nature of the interpolation. The interpolation is not over the complete set of points, but over a limited set, which is different for every particle.

The coefficient vector  $a(x)$  is found by minimising the following function:

$$J = \sum_j (p^t(x_j) \cdot a(x) - u_j)^2 W_j(x) \quad (\text{Eq. 2.16})$$

The solution is a standard procedure and will not be discussed here. The result is an interpolant of the form:

$$u^h(x) = \sum_j N_j(x) u_j \quad (\text{Eq. 2.17})$$

Where the shape function  $N_j$  takes the following form:

$$N_j(x) = p^t(x) \cdot A(x)^{-1} \cdot p(x_j) W_j(x), \quad (\text{Eq. 2.18})$$

with

$$A(x) = \sum_j p(x_j) p^t(x_j) W_j(x). \quad (\text{Eq. 2.19})$$

If the set of basis functions is simply:

$$p^t = [1] \quad (\text{Eq. 2.20})$$

then the MLS shape functions are:

$$N_j(x) = \frac{W_j(x)}{\sum_k W_k(x)}, \quad (\text{Eq. 2.21})$$

and one can recognise the Shepard functions that form the basis of normalised SPH.

#### 2.1.4 Zero-energy modes

Zero-energy modes are a problem that appears not only in particle methods. These spurious modes can also be found in the finite difference and finite element methods. Figure 2.1 illustrates this spurious mode for a field in 1D SPH. If one would approximate the derivative of the field shown in Figure 2.1 with a central difference formula:

$$\left. \frac{df}{dx} \right|_{x=x_i} = \frac{f(x_{i+1}) - f(x_{i-1}))}{x_{i+1} - x_{i-1}}, \quad (\text{Eq. 2.22})$$

then one would obtain:

$$\left. \frac{df}{dx} \right|_{x=x_i} = 0 \quad (\text{Eq. 2.23})$$

at all points. Hence this mode can not be detected, and can grow unhindered. This means that this mode could grow to a level where it dominates the solution.

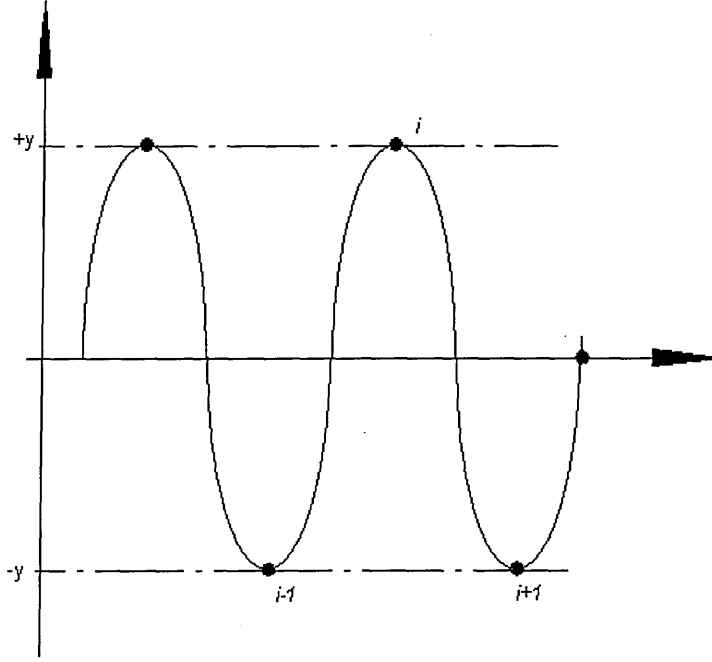


Figure 2.1: Zero Energy Modes.

### 2.1.5 Boundary Conditions

One of the main problems with meshless methods is the enforcement of boundary conditions. The approximations in SPH do not have the property of some other interpolants, such as FE shape functions, of the interpolated value of a variable at a particle being equal to the particle value of that variable i.e.

$$f^h(x_j) = \sum_I \Phi_I(x_j) f_I \neq f_j \quad (\text{Eq. 2.24})$$

Consequently it does not suffice to zero  $f_i$  at the boundary positions to enforce homogeneous boundary conditions. If the imposition of the free surface boundary condition (stress free condition) is simply ignored, then SPH will behave in an approximately correct manner, giving zero pressure for fluids and zero surface stresses for solids, because of the deficiency of particles at the boundary.

Campbell [ 20 ] made an early attempt to introduce a more systematic boundary condition treatment by re-considering the original kernel integral estimates and taking into account the boundary conditions through residual terms in the integral by parts. Randles and Libersky [ 77 ] were first to propose a more general treatment of the free surface boundary condition by using ghost particles to make up for the deficiency of particles on the boundary.

Belytschko et al [ 6 ] enforces the boundary conditions in his EFGM with a Lagrange multiplier method. This requires the solution of the following set of equations in the case of a static solid mechanics problem:

$$\begin{bmatrix} K & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} d \\ R \end{bmatrix} = \begin{bmatrix} f \\ q \end{bmatrix} \quad (\text{Eq. 2.25})$$

In this  $d$  are the displacements at the unconstrained nodes,  $R$  are the reaction forces at nodes with a displacement boundary condition.

Alternatively a penalty approach may be used in which case the following set of equations is obtained [ 92 ]:

$$(K + \alpha K^u) d = f + \alpha f^u, \quad (\text{Eq. 2.26})$$

where  $K^u$  and  $f^u$  are the extra terms due to the boundary conditions and  $\alpha$  is the penalty factor.

A third option is to use a transformation method that replaces the unknown variables with a new reduced set which implicitly satisfies the boundary conditions. This method is more efficient, and was first introduced by Chen [ 23 ]. There are two versions, a full transformation [ 23 ], and a boundary transformation method [ 24 ]. In this last version only the boundary and boundary influenced nodes are transformed. Boundary influenced nodes are those nodes that have boundary nodes as neighbours. An efficient boundary transformation algorithm based on d'Alembert's principle was proposed by Günther [ 37 ] and was applied to simulate compressible flow and fluid structure interaction problems. Starting with an original set of  $n$  equations with  $m$  constraints:

$$[M]\{\ddot{d}\} + [K]\{d\} - \{f\} = 0 \quad (n \text{ equations}) \quad (\text{Eq. 2.27})$$

$$G^T \{d\} = \{g\} \quad (m \text{ constraints}) \quad (\text{Eq. 2.28})$$

These equations are then transformed into an  $n-m$  set of equations with a new set of variables  $\{y\}$  such that the boundary conditions are satisfied:

$$\tilde{M}\ddot{y} + \tilde{K}y - \tilde{f} = 0 \quad (\text{Eq. 2.29})$$

After solving these equations for  $\{y\}$  the original unknown variables, the displacement vector  $\{d\}$ , can be obtained from:

$$\{d\} = J\{y\} + G\{g\} \quad (\text{Eq. 2.30})$$

Where  $J$  is the Jacobian matrix with:

$$J^T G = 0 \quad (\text{Eq. 2.31})$$

$$J^T J = I \quad (\text{Eq. 2.32})$$

Alternatively one can look at this method as transforming the shape functions. The interpolation function is:

$$u_i^h(x, t) = \sum_{j=1}^{NP} N_j(x) d_j(t) \quad (\text{Eq. 2.33})$$

By splitting the particles into two sets, one with boundary particles and one with non boundary particles, this equation can be written as:

$$u_i^h(x, t) = \sum_{j=1}^{NBP} N_j^b(x) d_j^b(t) + \sum_{j=1}^{NNBP} N_j^{nb}(x) d_j^{nb}(t) \quad (\text{Eq. 2.34})$$

These shape functions are then transformed into a new set:

$$u_i^h(x, t) = \sum_{j=1}^{NBP} \tilde{N}_j^b(x) u_j^b(t) + \sum_{j=1}^{NNBP} \tilde{N}_j^{nb}(x) d_j^{nb}(t) \quad (\text{Eq. 2.35})$$

These new shape functions satisfy the Kronecker delta property at the boundary.

The last method that can be used to enforce the essential boundary conditions is using a singular kernel for boundary nodes in order to partly restore the Kronecker delta property for the kernel [ 24 ]. It is based on the interpolating moving least squares interpolant by Lancaster and Salkauskas [ 54 ]. The method does not require the solution of an assembled system of equations, and the basic idea behind it is very simple. Starting from an original shape function:

$$N(x - x_j, h) \quad (\text{Eq. 2.36})$$

which in the case of SPH is equal to:

$$N(x - x_j, h) = \frac{m_j}{\rho_j} W(x - x_j, h) \quad (\text{Eq. 2.37})$$

the shape function is then modified as follows:

$$N'(x - x_j, h) = \begin{cases} \frac{N(x - x_j, h)}{|x - x_j|^p}, & \forall x_j \in \Gamma_u, \quad p > 0 \\ N(x - x_j, h), & \forall x_j \notin \Gamma_u \end{cases} \quad (\text{Eq. 2.38})$$

When the point  $x_j$  lies on the boundary the original shape function is divided by a function which is zero when  $x = x_j$ . The function used in Eq. 2.38,  $|x - x_j|^p$ , satisfies this criterion.



A new set of shape functions  $\{\tilde{N}'\}$  is then built as reproducing kernel functions, for example Shepard functions [Liu]:

$$\tilde{N}'(\mathbf{x} - \mathbf{x}_j, h) = \frac{N'(\mathbf{x} - \mathbf{x}_j, h)}{\sum_j N'(\mathbf{x} - \mathbf{x}_j, h)}. \quad (\text{Eq. 2.39})$$

It can be shown that these new shape functions possess the following property:

$$\tilde{N}'(\mathbf{x} - \mathbf{x}_j, h) \Big|_{\mathbf{x} \rightarrow \mathbf{x}_j} = 1, \quad j \in \Gamma_u \quad (\text{Eq. 2.40})$$

$$\tilde{N}'(\mathbf{x} - \mathbf{x}_i, h) \Big|_{\mathbf{x} \rightarrow \mathbf{x}_j} = 0, \quad j \in \Gamma_u \quad (\text{Eq. 2.41})$$

With this property the boundary conditions can be directly imposed. However the modified shape functions are not interpolation functions:

$$\tilde{N}'(\mathbf{x} - \mathbf{x}_j, h) \Big|_{\mathbf{x} \neq \mathbf{x}_j} \neq 0, \quad j \in \Gamma_u. \quad (\text{Eq. 2.42})$$

A final approach is to couple finite elements to the particles on the boundary and enforce the boundary conditions on the finite elements as the finite element shape functions satisfy the Kronecker delta property. However by doing this one loses the advantages of particle methods.

## 2.2 Galerkin Derivation of Discretised Momentum Equations

The aim of this section is to write a detailed Galerkin derivation of the SPH equations. This for the case of dynamics of continuous media. In a first stage the goal is to obtain a discretised set of equations

$$M\ddot{\mathbf{d}} + K\mathbf{d} - \mathbf{f} = 0, \quad (\text{Eq. 2.43a})$$

or by writing  $K\mathbf{d}$  as an internal force vector  $\mathbf{f}_i$

$$M\ddot{\mathbf{d}} + \mathbf{f}_i - \mathbf{f} = 0. \quad (\text{Eq. 2.43b})$$

as in the finite element method. In a second stage this will be extended to an explicit finite element method, using a central difference time integration.

The objective of this exercise is to compare the results with the existing central difference SPH algorithm currently used in the literature. This will point out any differences, and will allow new theory to be developed quicker as a Galerkin approach is very ‘procedural’. Also it could be a first step in the development of an implicit version of the Cranfield University SPH code. Having access to an implicit SPH code, or meshless code in general, would facilitate the testing of new capabilities.

For the purpose of this derivation the body force  $b$  and the boundary conditions will be ignored. This is an acceptable assumption since in the current SPH code these issues are not rigorously treated or are not present. Free surface boundary conditions are approximately satisfied due to the neighbour deficiency at the boundary. Displacement boundary conditions can be approximated by imposing them directly on the SPH points. However this derivation will prove useful when dealing with boundary condition treatment in SPH. Also this derivation can be used to derive an axisymmetric SPH formulation in a consistent manner.

The starting point of this derivation is the conservation of momentum equation:

$$\nabla \cdot \sigma + b = \rho a \quad \text{in } \Omega \quad (\text{Eq. 2.44})$$

with boundary conditions:

$$\sigma \cdot n = \bar{t} \quad \text{on } \Gamma_t \quad (\text{Eq. 2.45a})$$

$$\sigma \cdot n = 0 \quad \text{on } \Gamma_{to} \quad (\text{Eq. 2.45b})$$

$$u = \bar{u} \quad \text{on } \Gamma_u \quad (\text{Eq. 2.46})$$

where  $\Gamma_t \cup \Gamma_{to} \cup \Gamma_u = \Gamma$ , and  $\Gamma_t \cap \Gamma_{to} = \emptyset$ ,  $\Gamma_{to} \cap \Gamma_u = \emptyset$ ,  $\Gamma_t \cap \Gamma_u = \emptyset$ .

The first step is the derivation of a weak form of the conservation of momentum equation. In order to do this the equation is multiplied by a test function  $\delta w$  and integrated over the domain of the problem  $\Omega$ . This results in:

$$\int_{\Omega} \delta w \nabla \cdot \sigma \, dV + \int_{\Omega} \delta w b \, dV = \int_{\Omega} \delta w \rho a \, dV \quad (\text{Eq. 2.47})$$

Using the chain rule this becomes:

$$\int_{\Omega} \nabla \cdot (\delta w \sigma) \, dV - \int_{\Omega} \nabla \delta w \cdot \sigma \, dV + \int_{\Omega} \delta w b \, dV = \int_{\Omega} \delta w \rho a \, dV \quad (\text{Eq. 2.48})$$

Which using Green's theorem can be written as:

$$\int_{\Gamma} (\delta w \sigma) \cdot n \, d\Gamma - \int_{\Omega} \nabla \delta w \cdot \sigma \, dV + \int_{\Omega} \delta w b \, dV = \int_{\Omega} \delta w \rho a \, dV \quad (\text{Eq. 2.49})$$

The first term in this expression is equal to zero on  $\Gamma_{to}$ , so Eq. 2.48 can be rewritten as:

$$\int_{\Gamma_t \cup \Gamma_u} (\delta w \sigma) \cdot n \, d\Gamma - \int_{\Omega} \nabla \delta w \cdot \sigma \, dV + \int_{\Omega} \delta w b \, dV = \int_{\Omega} \delta w \rho a \, dV \quad (\text{Eq. 2.50})$$

The acceleration  $a$  can be replaced with  $\ddot{u}$ , the second time derivative of the displacements  $u$ :

$$\int_{\Omega} \delta w \rho \ddot{u} dV + \int_{\Omega} \nabla \delta w \cdot \sigma dV - \int_{\Omega} \delta w b dV - \int_{\Gamma_t \cup \Gamma_u} (\delta w \sigma) \cdot n d\Gamma = 0 \quad (\text{Eq. 2.51})$$

or in Voigt notation [ 7 ]:

$$\int_{\Omega} \rho \{\delta w\}^t \{\ddot{u}\} dV + \int_{\Omega} \{\nabla \delta w\}^t \{\sigma\} dV - \int_{\Omega} \{\delta w\}^t \{b\} dV - \int_{\Gamma_t \cup \Gamma_u} (\{\delta w\}^t \{\sigma\}) \cdot n d\Gamma = 0 \quad (\text{Eq. 2.52})$$

Eq. 2.52 can be discretised in the space domain by using shape functions for the displacement trial functions and the test functions:

$$\{\delta w\} = [N] \{\delta d\} \quad (\text{Eq. 2.53})$$

$$\{u\} = [N] \{d\} = \sum_j N_j d_j, \quad (\text{Eq. 2.54})$$

where  $[N]$  is an array of shape functions, and  $\{d\}$  is the nodal displacement vector. The precise shape and contents of the  $[N]$  array will be expanded upon later. The vectors  $\{d\}$  and  $\{\delta w\}$  are respectively the coordinates of a point and value of  $\delta w$  at a position in the domain  $\Omega$ .

Using these discretisations one can write the strains as

$$\{\epsilon\} = [\partial][N]\{d\} = [B]\{d\} \quad (\text{Eq. 2.55})$$

where  $[\partial]$  is a differential operator. The stresses for an elastic material can hence be written as follows:

$$\{\sigma\} = [E]\{\epsilon\} = [E][B]\{d\} = [E][\partial][N]\{d\} \quad (\text{Eq. 2.56})$$

and (see [ 93 ])

$$\{\nabla \delta w\}^T = \{\delta d\}^T [B]^T = \{\delta d\}^T [N]^T [\partial]^T \quad (\text{Eq. 2.57})$$

Finally,

$$\{\ddot{u}\} = [N]\{\ddot{d}\} \quad (\text{Eq. 2.58})$$

So the Galerkin expression becomes, by using Eq. 2.52, and Eq. 2.56, Eq. 2.57, Eq. 2.58:

$$\begin{aligned}
& \int_{\Omega} \{\delta d\}^T \rho [N]^T [N] \{\ddot{d}\} dV + \int_{\Omega} \{\delta d\}^T [B]^T \{\sigma\} dV \\
& - \int_{\Omega} \{\delta d\}^T [N]^T \{b\} dV - \int_{\Gamma_i \cup \Gamma_u} (\{\delta d\}^T [N]^T \{\sigma\}) \cdot n d\Gamma = 0
\end{aligned} \tag{Eq. 2.59}$$

or,

$$\begin{aligned}
& \int_{\Omega} \{\delta d\}^T \rho [N]^T [N] \{\ddot{d}\} dV + \int_{\Omega} \{\delta d\}^T [B]^T [E][B] \{d\} dV \\
& - \int_{\Omega} \{\delta d\}^T [N]^T \{b\} dV - \int_{\Gamma_i \cup \Gamma_u} (\{\delta d\}^T [N]^T \{\sigma\}) \cdot n d\Gamma = 0
\end{aligned} \tag{Eq. 2.60}$$

This equation can now be transformed in the familiar form by noting that  $\delta d$  and  $d$  are nodal values and hence can be taken out of the integral sign, resulting in:

$$\begin{aligned}
& \{\delta d\}^T \left( \int_{\Omega} \rho [N]^T [N] dV \{\ddot{d}\} + \int_{\Omega} [B]^T [E][B] dV \{d\} \right. \\
& \left. - \int_{\Omega} [N]^T \{b\} dV - \int_{\Gamma_i \cup \Gamma_u} ([N]^T \{\sigma\}) \cdot n d\Gamma \right) = 0
\end{aligned} \tag{Eq. 2.61}$$

This relation should hold for any test function  $\delta d$ , so one obtains:

$$[K]\{d\} + [M]\{\ddot{d}\} - \{f_b\} + \{f_e\} = 0 \tag{Eq. 2.62a}$$

$$[M]\{\ddot{d}\} + \{f_i\} - \{f_b\} + \{f_e\} = 0 \tag{Eq. 2.62b}$$

with

$$[K] = \int_{\Omega} [B]^T [E][B] dV \tag{Eq. 2.63}$$

$$[M] = \int_{\Omega} \rho [N]^T [N] dV \tag{Eq. 2.64}$$

If boundary conditions and body forces are ignored the last terms in Eq. 2.52 can be omitted:

$$\int_{\Omega} \rho \{\delta w\}^t \{\ddot{u}\} dV + \int_{\Omega} \{\nabla \delta w\}^t \{\sigma\} dV = 0 \tag{Eq. 2.65}$$

and one obtains

$$[M]\{\ddot{d}\} + \{f_i\} = 0 \quad (\text{Eq. 2.66})$$

So far the derivation has been identical to the finite element method, and no assumptions have been made about the shape functions, or the form of the [B] and [N] arrays. The shape functions can of course not be completely arbitrary. A general requirement on the shape functions is that they are part of a Sobolev space  $H^{m,s}$  [ 45 ], [ 28 ]. This means that the shape functions have to be part of a linear space with an inner product and Sobolev norm:

$$(u, v)_s = \int_{\Omega} \left( uv + u_i v_i + \dots + u_{\frac{i-1}{s}, k} v_{\frac{i-1}{s}, k} \right) d\Omega \quad (\text{Eq. 2.67})$$

$$\|u\|_s = (u, u)_s^{\frac{1}{2}} \quad (\text{Eq. 2.68})$$

defined, and finite.

This condition has implications for the continuity requirements of the shape functions and its derivatives [ 45 ] as, if  $s > n/2 + k$  then:

$$H^{m,s} \subset C_b^k, \quad (\text{Eq. 2.69})$$

where  $C_b^k$  is the space of functions with continuous and bounded derivatives of order up to  $k$ .

### 2.3 The general form of the [N] and [B] matrices in 3-D

In this paragraph the matrices [N] and [B] will be written out in component-form, so that their structure is clear. This information will be used in later paragraphs to write out the detailed formula to update the nodal accelerations. In 3-D the vector  $\{u\}$  will be 3x1 vector, the nodal displacement vector  $\{d\}$  will have dimensions  $3NP \times 1$ , and the shape function array [N] will have dimensions  $3 \times 3NP$ . So the displacement vector  $\{u\}$  can be expressed as follows:

$$\{u\} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} \dots & N_k & 0 & 0 & \dots \\ \dots & 0 & N_k & 0 & \dots \\ \dots & 0 & 0 & N_k & \dots \end{bmatrix} \begin{bmatrix} \vdots \\ d_{x_k} \\ d_{y_k} \\ d_{z_k} \\ \vdots \end{bmatrix} \quad (\text{Eq. 2.70})$$

Similarly the test function vector will have the following shape:

$$\{\delta w\} = \begin{bmatrix} \delta w_x \\ \delta w_y \\ \delta w_z \end{bmatrix} = \begin{bmatrix} \dots & N_k & 0 & 0 & \dots \\ \dots & 0 & N_k & 0 & \dots \\ \dots & 0 & 0 & N_k & \dots \end{bmatrix} \begin{bmatrix} \vdots \\ \delta d_{x_k} \\ \delta d_{y_k} \\ \delta d_{z_k} \\ \vdots \end{bmatrix} \quad (\text{Eq. 2.71})$$

The [B] array can be written as follows, and has dimensions 6x3NP:

$$\begin{aligned} [B] = [\partial][N] &= \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} \dots & N_k & 0 & 0 & \dots \\ \dots & 0 & N_k & 0 & \dots \\ \dots & 0 & 0 & N_k & \dots \end{bmatrix} \\ &= \begin{bmatrix} \dots & \frac{\partial N_k}{\partial x} & 0 & 0 & \dots \\ \dots & 0 & \frac{\partial N_k}{\partial y} & 0 & \dots \\ \dots & 0 & 0 & \frac{\partial N_k}{\partial z} & \dots \\ \dots & \frac{\partial N_k}{\partial y} & \frac{\partial N_k}{\partial x} & 0 & \dots \\ \dots & 0 & \frac{\partial N_k}{\partial z} & \frac{\partial N_k}{\partial y} & \dots \\ \dots & \frac{\partial N_k}{\partial z} & 0 & \frac{\partial N_k}{\partial x} & \dots \end{bmatrix} \end{aligned} \quad (\text{Eq. 2.72})$$

## 2.4 The general form of $[M]_{ij}$ and $[K]_{ij}$ in a 1-D SPH case

In a Finite Element context the shape functions would be defined such that the Kronecker-delta property is satisfied ie.

$$N_k(x_l) = \delta_{kl}, \quad (\text{Eq. 2.73})$$

using, for example, Lagrange polynomials [ 93 ]. In the context of the SPH method one starts from the following identity:

$$f(x) \equiv \int_{\Omega'} f(x') \delta(x - x') dx' \quad (\text{Eq. 2.74})$$

This is then approximated by replacing the Kronecker delta function with a, usually bell shaped, function  $W$ , the interpolation kernel, with compact support and satisfying the following conditions:

$$\int_{\Omega} W(x - x', h) dx' = 1 \quad (\text{Eq. 2.75})$$

And the kernel should act like the Kronecker delta if  $h$  goes to zero:

$$W(x - x', h) \xrightarrow{h \rightarrow 0} \delta(x - x') \quad (\text{Eq. 2.4})$$

Hence the above identity becomes:

$$f(x) \approx \int_{\Omega'} f(x') W(x - x', h) d\Omega' \quad (\text{Eq. 2.76})$$

Because the kernel  $W$  has compact support there will be a limited number of interpolation points interacting with  $x$ . The evaluation of the integral is then simply replaced by a sum over all the neighbouring particles within the support of  $x$ .

$$f(x) \approx \sum_{k=1}^{nnbr} \frac{m_k}{\rho_k} f(x_k) W(x - x_k, h) \quad (\text{Eq. 2.77})$$

In order to indicate this approximation triangular brackets will be put around  $f(x)$ :

$$\langle f(x) \rangle = \sum_{k=1}^{nnbr} \frac{m_k}{\rho_k} f(x_k) W(x - x_k, h) \quad (\text{Eq. 2.78})$$

This can now be written in terms of shape functions, by comparison with Eq. 2.54:

$$\langle f(x) \rangle = \sum_{k=1}^{nnbr} N_k(x) f(x_k) \quad (\text{Eq. 2.79})$$

So from this it can be concluded that the shape function in the case of an SPH formulation are of the following form:

$$N_k(x) = \frac{m_k}{\rho_k} W(x - x_k, h) \quad (\text{Eq. 2.80})$$

This expression can now be used, in combination with Eq. 2.63 and Eq. 2.64 to find a general expression for the stiffness and mass matrix elements. One can simply substitute the above equation into Eq. 2.53 and Eq. 2.54 to obtain, in a 1-D case:

$$\{u\} = \left[ \dots \frac{m_k}{\rho_k} W(x - x_k) \dots \right] \begin{bmatrix} \vdots \\ d_k \\ \vdots \end{bmatrix} \quad (\text{Eq. 2.81})$$

In 1-D the vector  $\{u\}$  will be a  $1 \times 1$  vector, the nodal displacement vector  $\{d\}$  will have dimensions  $NP \times 1$ , and the shape function array  $[N]$  will have dimensions  $1 \times NP$ . So the test function  $\{\delta u\}$  can be expressed as follows:

$$\{\delta w\} = \left[ \dots \frac{m_k}{\rho_k} W(x - x_k) \dots \right] \begin{bmatrix} \vdots \\ \delta d_k \\ \vdots \end{bmatrix} \quad (\text{Eq. 2.82})$$

The  $[B]$  array can be written as follows, and has dimensions  $1 \times NP$

$$[B] = [\partial][N] = \left[ \frac{\partial}{\partial x} \right] \left[ \dots N_k \dots \right] = \left[ \dots \frac{\partial N_k}{\partial x} \dots \right] \quad (\text{Eq. 2.83})$$

using Eq. 2.80 this becomes

$$[B] = \left[ \frac{\partial}{\partial x} \right] \left[ \dots \frac{m_k}{\rho_k} W(x - x_k) \dots \right] = \left[ \dots \frac{m_k}{\rho_k} \frac{\partial W(x - x_k)}{\partial x} \dots \right] \quad (\text{Eq. 2.84})$$

The constitutive matrix in a 1-D case is simply  $[E]$ . These expressions can now be substituted in Eq. 2.63 to give:

$$[K] = \int_{\Omega} [B]^T [E] [B] dV = \int_{\Omega} \begin{bmatrix} \vdots \\ \frac{\partial N_k}{\partial x} \\ \vdots \end{bmatrix} [E] \left[ \dots \frac{\partial N_l}{\partial x} \dots \right] dV \quad (\text{Eq. 2.85})$$

The  $kl$ -th term in this array is:

$$[K]_{kl} = \int_{\Omega} \frac{\partial N_k}{\partial x} E \frac{\partial N_l}{\partial x} dV \quad (\text{Eq. 2.86})$$

Using Eq. 2.80 to write the shape functions in terms of nodal masses and kernel function derivatives this becomes:



$$[K] = \int_{\Omega} \begin{bmatrix} \vdots \\ \frac{m_k}{\rho_k} \frac{\partial W(x-x_k)}{\partial x} \\ \vdots \end{bmatrix} [E] \begin{bmatrix} \dots & \frac{m_l}{\rho_l} \frac{\partial W(x-x_l)}{\partial x} & \dots \end{bmatrix} dV \quad (\text{Eq. 2.87})$$

$$[K]_{kl} = \int_{\Omega} \frac{m_k}{\rho_k} \frac{\partial W(x-x_k)}{\partial x} E \frac{m_l}{\rho_l} \frac{\partial W(x-x_l)}{\partial x} dV \quad (\text{Eq. 2.88})$$

Similarly the mass matrix can be written as:

$$[M] = \int_{\Omega} \rho [N]^T [N] dV = \int_{\Omega} \rho \begin{bmatrix} \vdots \\ N_k \\ \vdots \end{bmatrix} \begin{bmatrix} \dots & N_l & \dots \end{bmatrix} dV \quad (\text{Eq. 2.89})$$

The  $kl$ -th term in this array is:

$$[M]_{kl} = \int_{\Omega} \rho N_k N_l dV \quad (\text{Eq. 2.90})$$

Using Eq. 2.80 to write the shape functions in terms of nodal masses and kernel functions this becomes:

$$[M] = \int_{\Omega} \rho \begin{bmatrix} \vdots \\ \frac{m_k}{\rho_k} W(x-x_k) \\ \vdots \end{bmatrix} \begin{bmatrix} \dots & \frac{m_l}{\rho_l} W(x-x_l) & \dots \end{bmatrix} dV \quad (\text{Eq. 2.91})$$

$$[M]_{kl} = \int_{\Omega} \rho \frac{m_k}{\rho_k} W(x-x_k) \frac{m_l}{\rho_l} W(x-x_l) dV \quad (\text{Eq. 2.92})$$

These are the expressions that can be used to calculate the stiffness and mass matrices based on an SPH interpolation, rather than shape functions usually used in finite element methods. The last issue that is left open is the numerical integration of the integrals for every term in the mass and stiffness matrices. This will be addressed in paragraph 2.6.

If one imposes the condition that the approximation has to be able to reproduce the derivative of a linear deformation field exactly then this condition can be expressed as follows [ 9 ], [ 15 ]:

The deformation gradient can be written as follows:

$$F(x) = \sum_j d_j \otimes \tilde{\nabla} N_j(x), \quad (\text{Eq. 2.93})$$

where

$$\tilde{\nabla} N_j(x) = \frac{m_j}{\rho_j} \tilde{\nabla} W_j(x). \quad (\text{Eq. 2.94})$$

The  $\sim$  above the gradient operator  $\nabla$  is to indicate that a correction needs to be applied to the kernel gradient  $\nabla W$ . This is necessary because in the discretised form, by using the gradient of the unmodified kernel  $W$  it will not necessarily be possible to reproduce the gradient of a linear field exactly.

The modified kernel can be defined as a linear combination of the exact kernel derivatives:

$$\tilde{\nabla} N_j(x) = L(x) V_j \nabla W_j(x), \quad (\text{Eq. 2.95})$$

where  $L(x)$  is a correction matrix (dimensions: [ndim x ndim]).

If a the deformation gradient has to be exact for a linear displacement field then the following condition has to be imposed on the deformation gradient:

$$F(x_i) = \sum_j x_j \otimes \tilde{\nabla} N_j(x_i) = I, \quad (\text{Eq. 2.96})$$

where  $I$  is the identity matrix, and because the deformation field is linear the displacements can be replaced by the nodal coordinates  $x$ . This set of equations can then be solved for  $L(x)$  to obtain the corrected gradient functions.

A more general and more elegant way of obtaining this correction will be presented in the next section.

## 2.5 Derivation of Normalisation and Corrected Gradient Conditions

### 2.5.1 Introduction

The approximation of fields using a Normalised Corrected SPH (NCSPH) interpolation has been published in [ 87 ], [ 77 ], [ 13 ]. One of the main weaknesses in the way this Normalisation and Gradient Correction is derived in [ 13 ] is that they are based on global conservation of linear and angular momentum. This implies that the correction is only valid for a system of PDE's where these properties are applicable. Therefore, in the next sections of this paragraph, it will be demonstrated that the normalisation and gradient corrections can be derived by imposing homogeneity and isotropy of the solution space. This is a far more general condition as it does not restrict the PDE that is solved in the solution space.

### 2.5.2 Homogeneity of Space

Space is a linear vector field. The interpolation of the solution space itself should be independent of a translation of the coordinate axes. In other words, if one interpolates the coordinates of a point in space then the interpolated coordinates should be independent of the translation of coordinate axes.

In order to express this statement mathematically one can start by writing the general expression for the SPH interpolation of a vector field  $F$ :

$$\langle F(x) \rangle \Big|_{x=x_i} = \sum_j \frac{m_j}{\rho_j} F(x_j) W(x_i - x_j) \quad \text{Eq. 2.97}$$

If the field to be interpolated is the solution space then  $F = x$  and Eq. 2.97 becomes:

$$\langle x \rangle \Big|_{x=x_i} = \sum_j \frac{m_j}{\rho_j} x_j W(x_i - x_j) \quad \text{Eq. 2.98}$$

In a different, translated coordinate system, this equation is:

$$\langle x' \rangle \Big|_{x'=x'_i} = \sum_j \frac{m_j}{\rho_j} x'_j W(x'_i - x'_j) \quad \text{Eq. 2.99}$$

Where  $x'$  is the coordinate vector in the new coordinate system. If the translation vector by which the origin of the coordinate system was moved is defined as  $\Delta x$  then the relationship between  $x$  and  $x'$  is:

$$x' = x - \Delta x \quad \text{Eq. 2.100}$$

If the interpolated coordinates of a point are independent of the translation of coordinate axes then the following should hold:

$$\langle x' \rangle = \langle x \rangle - \Delta x. \quad \text{Eq. 2.101}$$

By substituting Eq. 2.100 into Eq. 2.99 for both  $x_i$  and  $x_j$  one obtains:

$$\langle x' \rangle = \sum_j \frac{m_j}{\rho_j} x_j W(x_i - x_j) - \sum_j \frac{m_j}{\rho_j} \Delta x W(x_i - x_j),$$

or

$$\langle x' \rangle = \langle x \rangle - \Delta x \sum_j \frac{m_j}{\rho_j} W(x_i - x_j). \quad \text{Eq. 2.102}$$

By comparison of Eq. 2.101 and Eq. 2.102 it is clear that the discretised space will only be homogeneous if:

$$\sum_j \frac{m_j}{\rho_j} W(x_i - x_j) = 1 \quad \text{Eq. 2.103}$$

### 2.5.3 Isotropy of Space

The SPH approximation should also be insensitive to a rotation of the coordinate axes (Isotropy of Space). The change in coordinates due to a rotation of the coordinate axes is (for small rotations):

$$x' = Cx$$

where  $C$  is the rotation matrix. For small rotations this can also be written as:

$$x' = x - \Delta\phi \times x, \quad \text{Eq. 2.104}$$

where  $\Delta\phi$  is the rotation vector.

If one wants to ensure that the SPH approximation does maintain the fact that space is isotropic then the approximation has to satisfy the following condition:

$$\langle x' \rangle \equiv \langle Cx \rangle = C \langle x \rangle$$

or

$$\langle C \rangle = C.$$

This means that the approximation to the rotation matrix has to be equal to the exact rotation matrix.

In order to develop this equation one can start by rewriting Eq. 2.104

$$\begin{aligned} x' &= x - \Delta\phi \times x \\ &= x - \nabla(\Delta\phi \times x) \cdot x \\ &= x - \phi^x x \\ &= (I - \phi^x) x \end{aligned}$$

where  $\Delta\phi^x$  is a skew-symmetric matrix:

$$\Delta\phi^x = \begin{bmatrix} 0 & -\Delta\phi_z & \Delta\phi_y \\ \Delta\phi_z & 0 & -\Delta\phi_x \\ -\Delta\phi_y & \Delta\phi_x & 0 \end{bmatrix}$$

This means that, for small rotations, the rotation matrix is given by:

$$C = I - \phi^x$$

The approximation of the rotated coordinates is:

$$\langle x' \rangle \equiv \langle Cx \rangle = \langle C \rangle \langle x \rangle = \langle I - \phi^x \rangle \langle x \rangle$$

This means that the requirement on the interpolation is:

$$I - \phi^x = \langle I - \phi^x \rangle,$$

or

$$\phi^x = \langle \phi^x \rangle$$

Expanding this expression leads to:

$$\begin{aligned} \langle \phi^x \rangle &= \sum_j \frac{m_j}{\rho_j} \Delta\phi \times x_j \nabla W(x_i - x_j) \\ &= \sum_j \frac{m_j}{\rho_j} (\phi^x x_j) \nabla W(x_i - x_j) \\ &= \phi^x \sum_j \frac{m_j}{\rho_j} x_j \nabla W(x_i - x_j) \end{aligned}$$

This means that

$$\phi^x = \langle \phi^x \rangle$$

if

$$\sum_j \frac{m_j}{\rho_j} x_j \nabla W(x_i - x_j) = I$$

## 2.6 Comparison of acceleration calculation with 'classical' SPH formulation

Now that the mass and stiffness matrices are known a comparison can be made between the results that have been obtained using the Galerkin derivation, and the 'classical' SPH equations.

In the Galerkin derivation, the first modification that needs to be made to the matrix equations to be able to use them in an explicit finite element manner is to use a lumped mass matrix. Because most of the lumping procedures are quite ad hoc, the different possibilities will not be reviewed and we will simply use the nodal mass  $m_i$  as the lumped mass. Hence the equation for the acceleration becomes:

$$m_i \langle a_i \rangle + \sum_{l=1}^{NP} \int_{\Omega} \frac{m_l}{\rho_l} \frac{\partial W(x - x_l)}{\partial x} E \frac{m_l}{\rho_l} \frac{\partial W(x - x_l)}{\partial x} dV d_l = 0 \quad (\text{Eq. 2.105})$$

which by taking out the factor  $m_i$  results in:

$$\langle a_i \rangle = - \sum_{l=1}^{NP} \int_{\Omega} \frac{1}{\rho_l} \frac{\partial W(x - x_l)}{\partial x} E \frac{m_l}{\rho_l} \frac{\partial W(x - x_l)}{\partial x} dV d_l \quad (\text{Eq. 2.106})$$

As mentioned before the numerical integration of the integral in the above expression still needs to be addressed. There are several options available. One could use a Gauss quadrature rule or other numerical integration methods. This is currently a topic of discussion in the meshless methods literature [ 66 ], [ 6 ], [ 14 ], [ 3 ], [ 26 ]. Several methods have been proposed but no method seems to offer the right combination of accuracy and computational efficiency. The simplest way to evaluate the integral is to simply use a weighted sum:

$$\int f(x') dV \approx \sum_j f(x_j) V_j = \sum_j f(x_j) \frac{m_j}{\rho_j} \quad (\text{Eq. 2.107})$$

which gives the final expression for the acceleration  $a_i$ :

$$\langle a_i \rangle = - \sum_{l=1}^{NP} \sum_{j=1}^{NP} \frac{1}{\rho_l} \frac{m_l}{\rho_l} \frac{m_j}{\rho_j} \frac{\partial W(x - x_l)}{\partial x} \bigg|_{x=x_j} \frac{\partial W(x - x_l)}{\partial x} \bigg|_{x=x_j} E d_l \quad (\text{Eq. 2.108})$$

using the fact that the kernel derivative is anti-symmetric:

$$\frac{\partial W(x_j - x_i)}{\partial x_j} = \frac{\partial W(x_i - x_j)}{\partial x_i} \quad (\text{Eq. 2.109})$$

$$\frac{\partial W(x_j - x_l)}{\partial x_j} = - \frac{\partial W(x_j - x_l)}{\partial x_l} \quad (\text{Eq. 2.110})$$

this expression can be rewritten as:

$$\langle a_i \rangle = \sum_{l=1}^{NP} \sum_{j=1}^{NP} \frac{1}{\rho_i} \frac{m_l}{\rho_l} \frac{m_j}{\rho_j} \frac{\partial W(x_i - x_j)}{\partial x_i} \frac{\partial W(x_j - x_l)}{\partial x_l} E d_l \quad (\text{Eq. 2.111})$$

Which is identical to the expression currently used in explicit time integration SPH formulations, cf. Eq. A.19.

In the finite element method the domain of integration  $\Omega$  is first split in elements with domain  $\Omega_e$ . The global integral is simply replaced by a sum of all the element integrals, and the shape functions become local shape functions [ 45 ]. In a particle method as has been described above this is not the case. The integral is evaluated over the whole domain, and the shape functions are global shape functions.

One conclusion of this comparison is that RKPM with Gauss integration or EFGM only differ from SPH in the evaluation of the integration of the weak form over the domain. By developing the SPH equations in this way it becomes clear where the different improvements on the SPH method act. For example the RKPM method changes the shape functions to restore consistency, but can still be used with a nodal integration. On the other hand one could use a background grid for the integration while still retaining the original SPH shape functions.

## 2.7 Explicit Formulation of the SPH Equations

In the case of an explicit formulation of the SPH equations one does not obtain a mass and stiffness matrix as in the previous paragraphs. Instead one integrates the equations in time, using for example a central difference scheme, in combination with a lumped mass matrix. The central difference method is commonly used as a time integration method in computational mechanics [ 7 ].

At the start of a timestep  $n$  the positions  $x$ , velocities  $v$ , stress  $\sigma$  and previous timestep  $\Delta t^{n-1/2}$  and time  $t^n$  are known. As illustrated in Figure 2.2 not all of the variables are known at the same time. The positions and density are known at the current problem time  $t^n$ , the stress is known at the time of the previous timestep  $t^{n-1}$ , while the velocities are known at the half timestep time  $t^{n-1/2}$ . The key to the central difference formula is the integration of variables from  $t^n$  to  $t^{n+1}$  using the derivative of the variable at the half time step  $t^{n+1/2}$ . In general the central difference formula for the first derivative of a function is:

$$\dot{f}^{n+1/2} = \frac{f^{n+1} - f^n}{\Delta t^{n+1/2}} = \frac{f^{n+1} - f^n}{t^{n+1} - t^n}, \quad (\text{Eq. 2.112})$$

which can be rearranged to an integration formula as follows:

$$f^{n+1} = f^n - \dot{f}^{n+1/2} \Delta t^{n+1/2}. \quad (\text{Eq. 2.113})$$

The first step in the time integration loop is to calculate the strain rates (see Figure 2.2). The gradient  $L$  and divergence of the velocity field are calculated as an intermediate step. In order to do this consistently the positions have to be moved back to the half step time  $t^{n-1/2}$  where the velocity field is known. From the velocity gradient the strain rates at  $t^{n-1/2}$  can be obtained. The divergence of the velocity field is used to advance the density to  $t^n$  (see Figure 2.2).

Once this is done, the stress can also be advanced to  $t^n$  by the use of a constitutive equation. The constitutive equation relates stress to strain or, in the case of non-linear material behaviour or time dependent problems, it is more convenient to relate the stress rate to the strain rate. The stress rate can then be integrated with the central difference formula (see Figure 2.2).

With the stress, density, and positions known at time  $t^n$  one can now evaluate the momentum equation to obtain the accelerations at time  $t^n$ . At this stage in the integration loop all variables are known at time  $t^n$ , apart from the velocity which is known at the timestep interval, so the time can be updated to  $t^n$  and the new timesteps  $\Delta t^n$  and  $\Delta t^{n+1/2}$  can be calculated. Because most variables are known at the same time this is usually the point where stateplots and time history output data is written.

The final steps in the time integration loop are the update of velocities and positions, using Eq. 2.113. All the variables have been advanced by one timestep, and the loop can be repeated.

In this integration loop two gradients are calculated, the velocity gradient and the stress gradient. These gradient calculations have to be approximated with SPH approximations. If a lumped mass matrix is used then the nodal accelerations are simply obtained by dividing the nodal force by the nodal mass. The nodal forces are evaluated on an element by element basis, so no stiffness matrix needs to be assembled. This, in combination with the use of a lumped mass matrix greatly reduces the computational effort.

In the next two paragraphs the SPH approximations of the strain-rate tensor (using the velocity gradient), and the acceleration vector are derived as they are used in an explicit time integration loop with lumped mass matrix.

Start of step  $n$

Know  $x^n, v^{n-1/2}, t^n, \Delta t^{n-1/2}, \sigma^{n-1}, \rho^{n-1}$

Store  $\Delta t^{n-1/2}$

Calculate strain rate using positions at half step:

$$x^{n-1/2} = x^n - \frac{1}{2} v^{n-1/2} \Delta t^{n-1/2}$$

$$L = (\nabla v)^T$$



$$\dot{\epsilon}^{n-1/2} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^T)$$

$$\dot{\epsilon}^{n-1/2} = [\mathbf{B}]\mathbf{v}$$

Update Density:

$$\rho^n = \rho^{n-1} + \dot{\rho}^{n-1/2} \Delta t^{n-1/2} = \rho^{n-1} + \nabla \cdot \mathbf{v}^{n-1/2} \Delta t^{n-1/2}$$

Calculate Stress using Constitutive Model:

$$\dot{\sigma}^{n-1/2} = f(\dot{\epsilon}^{n-1/2})$$

$$\sigma^n = \sigma^{n-1} + \dot{\sigma}^{n-1/2} \Delta t^{n-1/2}$$

Solve Momentum equation:

$$\mathbf{f}_i^n = \int [\mathbf{B}]^T \sigma^n d\Omega$$

$$\mathbf{M}\mathbf{a}^n + \mathbf{f}_i^n - \mathbf{f}_b^n + \mathbf{f}_c^n = 0$$

Update time,  $\Delta t$  using scale factor for stability:

$$\Delta t^{n+1/2} = \text{scale factor} \times \Delta t_{\text{crit}}^{n+1/2}$$

$$t^{n+1} = t^n + \Delta t^{n+1/2}$$

$$\Delta t^n = \frac{1}{2}(\Delta t^{n-1/2} + \Delta t^{n+1/2})$$

Update Velocity:

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \mathbf{a}^n \Delta t^n$$

Update Position:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1/2} \Delta t^{n+1/2}$$

Know  $\mathbf{x}^{n+1}, \mathbf{v}^{n+1/2}, t^{n+1}, \Delta t^{n+1/2}, \sigma^n, \rho^{n+1}$

End of step  $n$   $n = n + 1$

Figure 2.2: Central Difference Time Integration Loop.

### 2.7.1 The Conservation of Momentum Equation

The Galerkin procedure will be applied in the case of an explicit time integration scheme [ 7 ], [ 17 ], [ 61 ], [ 80 ]. Consequently there will be no stiffness or mass matrix, nor will there be a system of equations to be solved.

Starting from equation:

$$\mathbf{a} = \frac{1}{\rho} \nabla \cdot \boldsymbol{\sigma} = \nabla \cdot \frac{\boldsymbol{\sigma}}{\rho} + \frac{\boldsymbol{\sigma}}{\rho^2} \cdot \nabla \rho \quad (\text{Eq. 2.114})$$

We start by multiplying by a test function  $\delta w$  and integrating over the domain of the problem  $\Omega$  results in:

$$\int \delta w \mathbf{a} dV = \int \delta w \nabla \cdot \left( \frac{\boldsymbol{\sigma}}{\rho} \right) dV + \int \delta w \frac{1}{\rho^2} \nabla \rho \cdot \boldsymbol{\sigma} dV \quad (\text{Eq. 2.115})$$

The second term has to be linearised:

$$\int \delta w \mathbf{a} dV = \int \delta w \nabla \cdot \left( \frac{\boldsymbol{\sigma}}{\rho} \right) dV + \int \delta w \nabla \rho dV \cdot \frac{\boldsymbol{\sigma}_i}{\rho_i^2} \quad (\text{Eq. 2.116})$$

Now one can integrate by parts which results in:

$$\int \delta w \mathbf{a} dV = \int \nabla \cdot \left( \delta w \frac{\boldsymbol{\sigma}}{\rho} \right) dV + \frac{\boldsymbol{\sigma}_i}{\rho_i^2} \cdot \int \nabla (\delta w \rho) dV - \int \nabla \delta w \cdot \left( \frac{\boldsymbol{\sigma}}{\rho} \right) dV - \frac{\boldsymbol{\sigma}_i}{\rho_i^2} \cdot \int \rho \nabla \delta w dV$$

The first two integrals can be written as surface integrals by using Green's theorem:

$$\int \delta w \mathbf{a} dV = \int \left( \delta w \frac{\boldsymbol{\sigma}}{\rho} \right) \cdot \mathbf{n} d\Gamma + \frac{\boldsymbol{\sigma}_i}{\rho_i^2} \cdot \int (\delta w \rho) \mathbf{n} d\Gamma - \int \nabla \delta w \cdot \left( \frac{\boldsymbol{\sigma}}{\rho} \right) dV - \frac{\boldsymbol{\sigma}_i}{\rho_i^2} \cdot \int \rho \nabla \delta w dV$$

Because boundary conditions will be ignored the first two integrals will be assumed to be zero and will be left out. The acceleration can be written as the second time derivative of the displacement field, hence:

$$\int \delta w \ddot{\mathbf{u}} dV = - \int \nabla \delta w \cdot \left( \frac{\boldsymbol{\sigma}}{\rho} \right) dV - \frac{\boldsymbol{\sigma}_i}{\rho_i^2} \cdot \int \rho \nabla \delta w dV \quad (\text{Eq. 2.117})$$

Now the test function and the trial function can be discretised using shape functions:

$$\{\delta w\} = [\mathbf{N}] \{\delta d\} \quad (\text{Eq. 2.118})$$

$$\{\mathbf{u}\} = [\mathbf{N}] \{\mathbf{d}\} \Rightarrow \{\ddot{\mathbf{u}}\} = [\mathbf{N}] \{\ddot{\mathbf{d}}\} \quad (\text{Eq. 2.119})$$

The gradient of these vectors is:

$$\{\nabla \delta w\} = [\partial][N]\{\delta d\} = [B]\{\delta d\} \quad (\text{Eq. 2.120})$$

In component form  $[N]$ ,  $[\delta]$  and  $[B]$  are  $[45]$ ,  $[93]$ :

$$[N] = \begin{bmatrix} \dots & N_j & 0 & \dots \\ \dots & 0 & N_j & \dots \end{bmatrix} \quad (\text{Eq. 2.121})$$

$$[\delta] = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}, \quad (\text{Eq. 2.122})$$

and by combining these two expressions,

$$[B] = \begin{bmatrix} \dots & \frac{\partial N_j}{\partial x} & 0 & \dots \\ \dots & 0 & \frac{\partial N_j}{\partial y} & \dots \\ \dots & \frac{\partial N_j}{\partial y} & \frac{\partial N_j}{\partial x} & \dots \end{bmatrix} \quad (\text{Eq. 2.123})$$

Substituting these equations in Eq. 2.117 results in:

$$\int \{\delta d\}^T [N]^T [N] \{\ddot{u}\} dV = - \int \{\delta d\}^T [B]^T \cdot \left( \frac{\sigma}{\rho} \right) dV - \int \rho \{\delta d\}^T [B]^T dV \cdot \frac{\sigma_i}{\rho_i^2} \quad (\text{Eq. 2.124})$$

This equation is valid for any  $\{\delta d\}$  so this term can be deleted. This then gives the final relation for the momentum equation in Voigt notation:

$$\int [N]^T [N] dV \{\ddot{u}\} = - \int [B]^T \cdot \frac{\sigma}{\rho} dV - \int \rho [B]^T dV \cdot \frac{\sigma_i}{\rho_i^2} \quad (\text{Eq. 2.125})$$

In order to write out these equations in component form the two terms on the RHS of the equations will be developed separately.

The first term on the RHS of Eq. 2.125 is:

$$- \int [B]^T \cdot \frac{\sigma}{\rho} dV \quad (\text{Eq. 2.126})$$

By substitution of Eq. 2.123 and

$$\left\{ \frac{\sigma}{\rho} \right\} = \begin{Bmatrix} \frac{\sigma_{xx}}{\rho} \\ \frac{\sigma_{yy}}{\rho} \\ \frac{\sigma_{xy}}{\rho} \end{Bmatrix} \quad (\text{Eq. 2.127})$$

this first term can be rewritten as:

$$- \int \begin{bmatrix} \vdots \\ \frac{\partial N_i}{\partial x} \frac{\sigma_{xx}}{\rho} + \frac{\partial N_i}{\partial y} \frac{\sigma_{xy}}{\rho} \\ \frac{\partial N_i}{\partial x} \frac{\sigma_{xy}}{\rho} + \frac{\partial N_i}{\partial y} \frac{\sigma_{yy}}{\rho} \\ \vdots \end{bmatrix} dV \quad (\text{Eq. 2.128})$$

The next step is to substitute the correct expressions for the shape functions. In the case of normalised shape functions these are:

$$N_k = \frac{m_k}{\rho_k} \frac{W(x - x_k)}{\sum_k \frac{m_k}{\rho_k} W(x - x_k)} = \frac{m_k}{\rho_k} \frac{W(x - x_k)}{\sum_k V_k W_k} \quad (\text{Eq. 2.129})$$

In order to improve the clarity of subsequent equations the following notation will be used:

$$N_k = \frac{m_k}{\rho_k} \frac{W(x - x_k)}{\sum_k V_k W_k} = \frac{m_k}{\rho_k} \tilde{W}(x - x_k) \quad (\text{Eq. 2.130})$$

and the gradient of this normalised kernel is:

$$\nabla \tilde{W}(x - x_k) = \frac{\nabla W(x - x_k) \sum_k V_k W_k - W(x - x_k) \sum_k V_k \nabla W_k}{\left( \sum_k V_k W_k \right)^2} \quad (\text{Eq. 2.131})$$

$$\begin{bmatrix} \frac{\partial \tilde{W}}{\partial x} \\ \frac{\partial \tilde{W}}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\frac{\partial W_k}{\partial x} \sum_k V_k W_k - W_k \sum_k V_k \frac{\partial W_k}{\partial x}}{\left( \sum_k V_k W_k \right)^2} \\ \frac{\frac{\partial W_k}{\partial y} \sum_k V_k W_k - W_k \sum_k V_k \frac{\partial W_k}{\partial y}}{\left( \sum_k V_k W_k \right)^2} \end{bmatrix} \quad (\text{Eq. 2.132})$$

and the first term on the RHS of the momentum equation can be written as:

$$- \int \frac{m_i}{\rho_i} \begin{bmatrix} \vdots \\ \frac{\partial \tilde{W}(x-x_i)}{\partial x} \frac{\sigma_{xx}}{\rho} + \frac{\partial \tilde{W}(x-x_i)}{\partial y} \frac{\sigma_{xy}}{\rho} \\ \frac{\partial \tilde{W}(x-x_i)}{\partial x} \frac{\sigma_{xy}}{\rho} + \frac{\partial \tilde{W}(x-x_i)}{\partial y} \frac{\sigma_{yy}}{\rho} \\ \vdots \end{bmatrix} dV \quad (\text{Eq. 2.133})$$

Now the integral can be approximated by a sum. The integration is over the complete domain:

$$- \int_{x=y} \frac{m_i}{\rho_i} \begin{bmatrix} \vdots \\ \frac{\partial \tilde{W}(x-x_i)}{\partial x} \frac{\sigma_{xx}}{\rho} + \frac{\partial \tilde{W}(x-x_i)}{\partial z} \frac{\sigma_{xy}}{\rho} \\ \frac{\partial \tilde{W}(x-x_i)}{\partial x} \frac{\sigma_{xy}}{\rho} + \frac{\partial \tilde{W}(x-x_i)}{\partial y} \frac{\sigma_{yy}}{\rho} \\ \vdots \end{bmatrix} dx dy \quad (\text{Eq. 2.134})$$

The integral over  $dx dy$  is again replaced by a sum.

$$- \sum_{j=1}^{NP} \frac{m_j}{\rho_j} \frac{m_i}{\rho_i} \begin{bmatrix} \vdots \\ \frac{\partial \tilde{W}(x_j-x_i)}{\partial x_j} \frac{\sigma_{xx,j}}{\rho_j} + \frac{\partial \tilde{W}(x_j-x_i)}{\partial y_j} \frac{\sigma_{xy,j}}{\rho_j} \\ \frac{\partial \tilde{W}(x_j-x_i)}{\partial x_j} \frac{\sigma_{xy,j}}{\rho_j} + \frac{\partial \tilde{W}(x_j-x_i)}{\partial y_j} \frac{\sigma_{yy,j}}{\rho_j} \\ \vdots \end{bmatrix} \quad (\text{Eq. 2.135})$$

The final manipulations of this expression are using the fact that:

$$\nabla_j W = -\nabla_i W. \quad (\text{Eq. 2.136})$$

This results in:

$$\sum_{j=1}^{NP} \frac{m_j}{\rho_j} \frac{m_i}{\rho_i} \left[ \begin{array}{c} \vdots \\ \frac{\partial \tilde{W}(x_j - x_i)}{\partial x_i} \frac{\sigma_{xx,j}}{\rho_j} + \frac{\partial \tilde{W}(x_j - x_i)}{\partial y_i} \frac{\sigma_{xy,j}}{\rho_j} \\ \frac{\partial \tilde{W}(x_j - x_i)}{\partial x_i} \frac{\sigma_{xy,j}}{\rho_j} + \frac{\partial \tilde{W}(x_j - x_i)}{\partial y_i} \frac{\sigma_{yy,j}}{\rho_j} \\ \vdots \end{array} \right] \quad (\text{Eq. 2.137})$$

The same approach will now be applied to the second term of the RHS of the momentum equation which is:

$$- \int \rho [B]^T dV \cdot \left\{ \frac{\sigma_i}{\rho_i^2} \right\} \quad (\text{Eq. 2.138})$$

By substituting in Eq. 2.123 for [B] this results in:

$$- \int \rho \left[ \begin{array}{ccc} \vdots & \vdots & \vdots \\ \frac{\partial N_i}{\partial x} & 0 & \frac{\partial N_i}{\partial y} \\ 0 & \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \\ \vdots & \vdots & \vdots \end{array} \right] dV \cdot \left\{ \frac{\sigma_i}{\rho_i^2} \right\} \quad (\text{Eq. 2.139})$$

After multiplying out this gives:

$$- \int \rho \left[ \begin{array}{c} \vdots \\ \frac{\partial N_i}{\partial x} \frac{\sigma_{xx,i}}{\rho_i^2} + \frac{\partial N_i}{\partial y} \frac{\sigma_{xy,i}}{\rho_i^2} \\ \frac{\partial N_i}{\partial y} \frac{\sigma_{xy,i}}{\rho_i^2} + \frac{\partial N_i}{\partial x} \frac{\sigma_{yy,i}}{\rho_i^2} \\ \vdots \end{array} \right] dV \quad (\text{Eq. 2.140})$$

Replacing  $N_i$  with Eq. 2.130 results in:

$$- \int \rho \frac{m_i}{\rho_i} \left[ \begin{array}{c} \vdots \\ \frac{\partial \tilde{W}(x - x_i)}{\partial x} \frac{\sigma_{xx,i}}{\rho_i^2} + \frac{\partial \tilde{W}(x - x_i)}{\partial y} \frac{\sigma_{xy,i}}{\rho_i^2} \\ \frac{\partial \tilde{W}(x - x_i)}{\partial y} \frac{\sigma_{xy,i}}{\rho_i^2} + \frac{\partial \tilde{W}(x - x_i)}{\partial x} \frac{\sigma_{yy,i}}{\rho_i^2} \\ \vdots \end{array} \right] dV \quad (\text{Eq. 2.141})$$

Again, one can now replace the integral with a sum;

$$-\sum_{j=1}^{NP} \frac{m_j}{\rho_j} \frac{m_i}{\rho_i} \rho_j \begin{bmatrix} \frac{\partial \tilde{W}(x_j - x_i)}{\partial x_j} \frac{\sigma_{xx,i}}{\rho_i^2} + \frac{\partial \tilde{W}(x_j - x_i)}{\partial y_j} \frac{\sigma_{xy,i}}{\rho_i^2} \\ \frac{\partial \tilde{W}(x_j - x_i)}{\partial y_j} \frac{\sigma_{xy,i}}{\rho_i^2} + \frac{\partial \tilde{W}(x_j - x_i)}{\partial x_j} \frac{\sigma_{yy,i}}{\rho_i^2} \end{bmatrix} \quad (\text{Eq. 2.142})$$

Finally one can use Eq. 2.136, which gives the final expression for the second term on the RHS of the momentum equation:

$$\sum_{j=1}^{NP} m_j \frac{m_i}{\rho_i} \begin{bmatrix} \frac{\partial \tilde{W}(x_j - x_i)}{\partial x_i} \frac{\sigma_{xx,i}}{\rho_i^2} + \frac{\partial \tilde{W}(x_j - x_i)}{\partial y_i} \frac{\sigma_{xy,i}}{\rho_i^2} \\ \frac{\partial \tilde{W}(x_j - x_i)}{\partial y_i} \frac{\sigma_{xy,i}}{\rho_i^2} + \frac{\partial \tilde{W}(x_j - x_i)}{\partial x_i} \frac{\sigma_{yy,i}}{\rho_i^2} \end{bmatrix} \quad (\text{Eq. 2.143})$$

These two expressions can now be combined to give the SPH equation for the momentum equation. If we assume a lumped mass matrix is used then the  $\frac{m_i}{\rho_i}$  term cancels with the one on the LHS. For clarity purposes the stress and kernel derivative terms can be written as a matrix product:

$$\begin{bmatrix} \vdots \\ a_{r,i} \\ a_{z,i} \\ \vdots \end{bmatrix} = \sum_{j=1}^{NP} m_j \begin{bmatrix} \frac{\sigma_{xx,j}}{\rho_j^2} + \frac{\sigma_{xx,i}}{\rho_i^2} & \frac{\sigma_{xy,j}}{\rho_j^2} + \frac{\sigma_{xy,i}}{\rho_i^2} \\ \frac{\sigma_{xy,j}}{\rho_j^2} + \frac{\sigma_{xy,i}}{\rho_i^2} & \frac{\sigma_{yy,j}}{\rho_j^2} + \frac{\sigma_{yy,i}}{\rho_i^2} \\ \vdots & \vdots \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{W}(x_j - x_i)}{\partial x_i} \\ \frac{\partial \tilde{W}(x_j - x_i)}{\partial y_i} \end{bmatrix} \quad (\text{Eq. 2.144})$$

### 2.7.2 Strain Rate

The strain rate for a certain particle is calculated using:

$$\{\dot{\epsilon}\}_i = [B]\{v\} \quad (\text{Eq. 2.145})$$

$$\begin{Bmatrix} \dot{\epsilon}_{xx} \\ \dot{\epsilon}_{yy} \\ \dot{\gamma}_{xy} \end{Bmatrix} = \begin{bmatrix} \dots & \frac{\partial N_j}{\partial x} & 0 & \dots \\ \dots & 0 & \frac{\partial N_j}{\partial y} & \dots \\ \dots & \frac{\partial N_j}{\partial y} & \frac{\partial N_j}{\partial x} & \dots \end{bmatrix} \begin{Bmatrix} \vdots \\ v_{x,j} \\ v_{y,j} \\ \vdots \end{Bmatrix} \quad (\text{Eq. 2.146})$$

The ‘engineering’ zr-strain is here used instead of the tensor strain. Written out in full and replacing the shape function with Eq. 2.130 results in:

$$\dot{\epsilon}_{xx} = \sum_{j=1}^{NP} \frac{m_j}{\rho_j} \frac{\partial \tilde{W}(x - x_j)}{\partial x} v_{x,j} \quad (\text{Eq. 2.147})$$

$$\dot{\epsilon}_{yy} = \sum_{j=1}^{NP} \frac{m_j}{\rho_j} \frac{\partial \tilde{W}(x - x_j)}{\partial y} v_{y,j} \quad (\text{Eq. 2.148})$$

$$\dot{\gamma}_{xy} = \sum_{j=1}^{NP} \frac{m_j}{\rho_j} \left( \frac{\partial \tilde{W}(x - x_j)}{\partial y} v_{x,j} + \frac{\partial \tilde{W}(x - x_j)}{\partial x} v_{y,j} \right) \quad (\text{Eq. 2.149})$$

Evaluating the strain rates at particle i one obtains:

$$\dot{\epsilon}_{xx,i} = \sum_{j=1}^{NP} \frac{m_j}{\rho_j} \frac{\partial \tilde{W}(x_i - x_j)}{\partial x_i} v_{x,j} \quad (\text{Eq. 2.150})$$

$$\dot{\epsilon}_{yy,i} = \sum_{j=1}^{NP} \frac{m_j}{\rho_j} \frac{\partial \tilde{W}(x_i - x_j)}{\partial y_i} v_{y,j} \quad (\text{Eq. 2.151})$$

$$\dot{\gamma}_{xy,i} = \sum_{j=1}^{NP} \frac{m_j}{\rho_j} \left( \frac{\partial \tilde{W}(x_i - x_j)}{\partial y_i} v_{x,j} + \frac{\partial \tilde{W}(x_i - x_j)}{\partial x_i} v_{y,j} \right) \quad (\text{Eq. 2.152})$$



## Chapter 3 Contact Modelling in Meshless Methods

### 3.1 Introduction

The analysis of contact problems raises great interest in several fields of research: metal forming, crashworthiness, projectile impact, post-buckling response of structures. When modelling water impact it is, of course, essential to be able to model the contact between the impacting structure and the water accurately. The main problem in modelling water impact is that the volume of water will drastically change shape during the impact. Therefore it is essential to have discretisation method that allows large deformations to be modelled easily, and also to have a robust contact algorithm. When using particles to model water it would be very difficult to define contact surfaces for the water in a traditional manner such as used in FE. An efficient contact algorithm should be able to automatically update the interface geometry as the domain of the water deforms. If one had to construct a surface around the SPH domain to consider contact then the benefit of using meshless methods for a large number of impact problems would be significantly reduced.

The frictional forces between an impacting structure and water are very small. Therefore, friction has been ignored in the developed contact algorithm and in the remainder of this chapter only frictionless contact is considered.

#### 3.1.1 Kuhn-Tucker Conditions

Before giving an account of modern techniques to evaluate the contact in Explicit Finite Element codes, it is useful to note that the contact from a mathematical point of view is a constraint on the solution. The mathematical expressions for the constraints on the solution by the contact condition are known as the Kuhn-Tucker Conditions.

In order to express these conditions mathematically we introduce a gap function,  $g$ , defined as the interpenetration distance between the two bodies,  $\Omega_I$  and  $\Omega_J$ . Hence, the gap function depends on the position of the both bodies, say  $\mathbf{x}^{(I)}$  and  $\mathbf{x}^{(J)}$ :

$$g = g(\mathbf{x}^{(I)}, \mathbf{x}^{(J)}) = g(\|\mathbf{x}^{(J)} - \mathbf{x}^{(I)}\|) \quad (\text{Eq. 3.1})$$

Because we are limiting our attention to the problem of frictionless contact, the only component of the contact force of interest is the component normal to the body,  $\mathbf{t}_N$ . The Kuhn-Tucker Conditions can be written as:

$$g \leq 0 \quad (\text{Eq. 3.2a})$$

$$\mathbf{t}_n \geq 0 \quad (\text{Eq. 3.2b})$$

$$\mathbf{t}_n \cdot \mathbf{g} = 0 \quad (\text{Eq. 3.2c})$$

These conditions express the impenetrability, the compressive normal traction constraint, and the fact that the traction is only non-zero when  $g = 0$ .

If the rate of change of the gap function is written as  $\dot{g}$ , then the persistency of the contact, which by some authors is called Unitary Contact Condition, can be expressed as:

$$\mathbf{t}_n \cdot \dot{\mathbf{g}} = 0 \quad (\text{Eq. 3.2d})$$

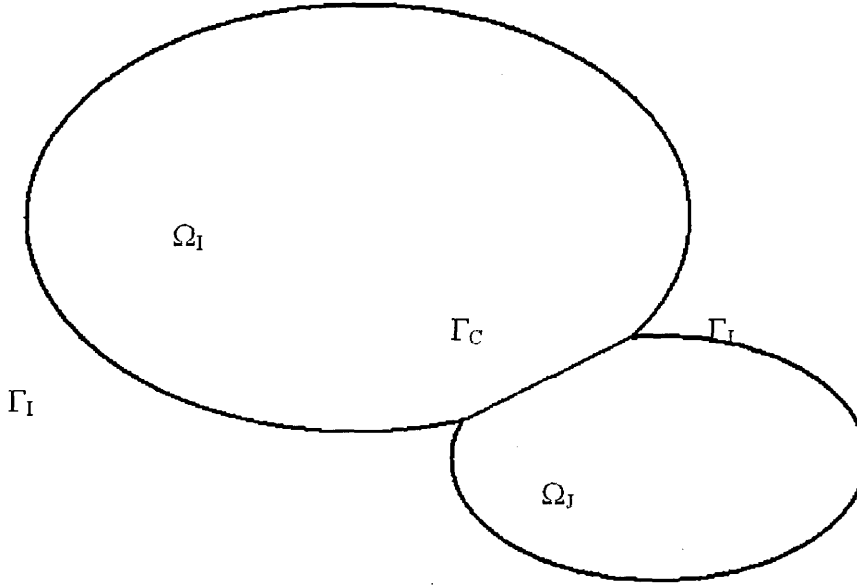


Figure 3.1: Schematic representation of two bodies in contact.

If the equations of motion are expressed in the variational rate form the contact constraint used to impose the contact is:

$$\delta G = \delta G(\mathbf{t}_n \cdot \dot{\mathbf{g}}) = 0 \quad (\text{Eq. 3.3})$$

Where  $\delta G$  is a (scalar) function of contact persistency. This equation expresses the fact that the normal components of the contact forces do no work. Which is clear by noting that if the body is in contact and remains in contact then  $\dot{\mathbf{g}} = 0$ , if the bodies are not in contact then  $\mathbf{t}_n = 0$ .

The contact surface  $\Gamma_C$  is:

$$\Gamma_C = \partial\Omega^{(I)} \cap \partial\Omega^{(J)}.$$

This contact surface is shared by both the bodies in contact and the following relationships hold:

$$\partial\Omega^{(i)} = \Gamma_D^{(i)} \cup \Gamma_N^{(i)} \cup \Gamma_C \quad \Gamma_D^{(i)} \cap \Gamma_C = 0 \quad \Gamma_N^{(i)} \cap \Gamma_C = 0$$

and

$$\partial\Omega^{(j)} = \Gamma_D^{(j)} \cup \Gamma_N^{(j)} \cup \Gamma_C \quad \Gamma_D^{(j)} \cap \Gamma_C = 0 \quad \Gamma_N^{(j)} \cap \Gamma_C = 0.$$

in this  $\Gamma_D$  and  $\Gamma_N$  are the sections of the boundary with Dirichlet and Von Neumann boundary conditions respectively. Note that the normal,  $n$ , used to define the normal components of forces and displacements was also implicitly assumed to be the same for both bodies.

The most widely used techniques for the enforcement of the contact condition in modern explicit finite element codes are the Lagrange multipliers, the penalty method, the kinematical constraint method and the distributed parameters method. Following a detailed consideration of these methods the penalty method has been chosen as the method for contact enforcement in the new contact algorithm because of its simplicity and numerical efficiency.

### 3.1.2 General Variational Framework and Conservation of Energy and Momentum

Several studies and investigations have been carried out in order to achieve a general framework from which to derive a (variational) formulation for the contact problem, which would provide the necessary general background for contact modelling independent from the type of spatial discretisation [ 81 ].

To describe the general case of motion of non-linear continua the equations of conservation of mass, conservation of linear momentum, the constitutive equation and the thermodynamic second law have to be integrated in time and space.

If the external loads are conservative and thermal effects do not affect the constitutive law, total energy is conserved and the energy conservation equation does not have to be integrated explicitly. In this case the dynamics of the continuum is completely described through the conservation of the linear  $L$  and angular  $J$  momentum:

$$\frac{d}{dt} L_t = 0 \quad (\text{Eq. 3.4a})$$

and

$$\frac{d}{dt} J_t = 0 \quad (\text{Eq. 3.4b})$$

These equations are usually used in the consideration of orbiting structures or in fluid dynamics.

From the conservation of energy equation one can write:

$$\frac{d}{dt} E_t^{\text{tot}} = P_t^{\text{ext}} - D_t^{\text{int}}$$

It follows that, in the absence of external loads and referring to an arbitrary constitutive law:

$$\frac{d}{dt} E_t^{\text{tot}} = -D_t^{\text{int}} \quad (\text{Eq. 3.4c})$$

Where  $D_t^{\text{int}}$  is the dissipation, which eventually depends on the non-linearity of the material, and as usual,  $E_t^{\text{tot}} = E_t^{\text{kin}} + E_t^{\text{int}}$ . For an incompressible fluid  $E_t^{\text{int}} = 0$ . While for an elastic material  $E_t^{\text{int}} = \Pi^{\text{int}}$ , where  $\Pi^{\text{int}}$  is the elastic potential energy.

In the case of a discrete form of the system described above, Eq. 3.4 becomes:

$$L_{n+1} = L_n \quad (\text{Eq. 3.5a})$$

$$J_{n+1} = J_n \quad (\text{Eq. 3.5b})$$

Supplemented with the constraint:

$$E_{n+1}^{\text{tot}} < E_n^{\text{tot}} \quad (\text{Eq. 3.5c})$$

Where

$$E_{n+1}^{\text{tot}} \equiv E_n^{\text{tot}} \Leftrightarrow D_n^{\text{int}} \equiv 0$$

Simo and Tarnow [ 81 ] demonstrated that the discrete form of the equations does not necessarily inherit the same conservative characteristics, which independently from the constitutive law, are intrinsic in the problem analysed. If the conservative nature of equations is preserved the scheme is called energy-momentum conserving.

It was demonstrated that independently from the numerical scheme used to perform the integration in time, linear and angular momentum are conserved if the configuration at half time step, i.e.  $t_{n+\frac{1}{2}}$ , is considered as an intermediate configuration. In fact, referring to an arbitrary configuration at time  $t_{n+\alpha}$  (see Figure 3.2), the following holds:

$$L_{n+1} - L_n = \Delta t F_{n+\alpha}^{\text{ext}} \quad \forall \alpha, \alpha \in [0,1] \quad (\text{Eq. 3.6a})$$

$$J_{n+1} - J_n = \Delta t M_{n+\alpha}^{\text{ext}} \quad \alpha = 1/2 \quad (\text{Eq. 3.6b})$$

Where  $F_{n+\alpha}^{\text{ext}}$  and  $M_{n+\alpha}^{\text{ext}}$  are respectively the resultant of the external force and moment defined at intermediate time  $t_{n+\alpha}$ . Noticing that in the absence of external loads, the same conditions are identically satisfied for any  $\alpha$ .

Unfortunately, the conservation of linear and angular momentum is not sufficient for the consistency of the numerical method since, under the condition expressed above, the conservation of linear and angular momentum does not depend on the way the constitutive equation is evaluated. Hence, the conservation of linear and angular momentum does not necessarily satisfy the dissipation constraint, which on the contrary depends explicitly on the constitutive law. In order to provide overall consistency to a method the total energy conservation condition has to be satisfied, as well.

This statement can be demonstrated referring to the weak form of the principle of the virtual work. In particular, in order to ensure conservation of energy it is necessary to define the intermediate configuration such that it does not produce non-physical stresses, for details see [ 81 ]. In spite of the apparent difficulty in finding such a configuration, it exists and it is the configuration that corresponds to  $\alpha = 1/2$ .

In the absence of external loads and in the most general case of a non-linear constitutive law, the configuration to be used to evaluate the constitutive equation is not longer the one at half time step but the one consistent with the Second Law of Thermodynamics. And, the need to use the proper intermediate configuration deeply alters the approach to the time integration since it requires the solution of an implicit equation at each time step.

The time integration scheme modified according to the results discussed above and coupled with a finite element spatial discretisation, initially proposed by [ 55 ], has been applied to solve specific contact problems. The approach demonstrated the merit of having the correct time integration for the correct treatment of contact. This work provided the necessary background for further investigations.

The work published by [ 56 ] represented in some respect a backward step in comparison with the scheme described above. The approach followed was the same, but dealt with a linear-elastic material and the contact was frictionless. These assumptions were the basis for a fully conservative implicit scheme, which eventually led to some important conclusions. Concerning the time integration, it was pointed out that an implicit scheme allows for damping of the high-frequency oscillations related to the spatial and temporal discretisation. And, the need for a method able to embody the dynamics of the contact and at the same time satisfy the Kuhn-Tucker conditions was identified.

Unfortunately, mathematically these conditions are not so simple to enforce. In particular, the conditions to be imposed on the contact surface are neither Dirichlet nor von Neumann since both displacement and stress are unknown but related to each other through the Kuhn-Tucker conditions.

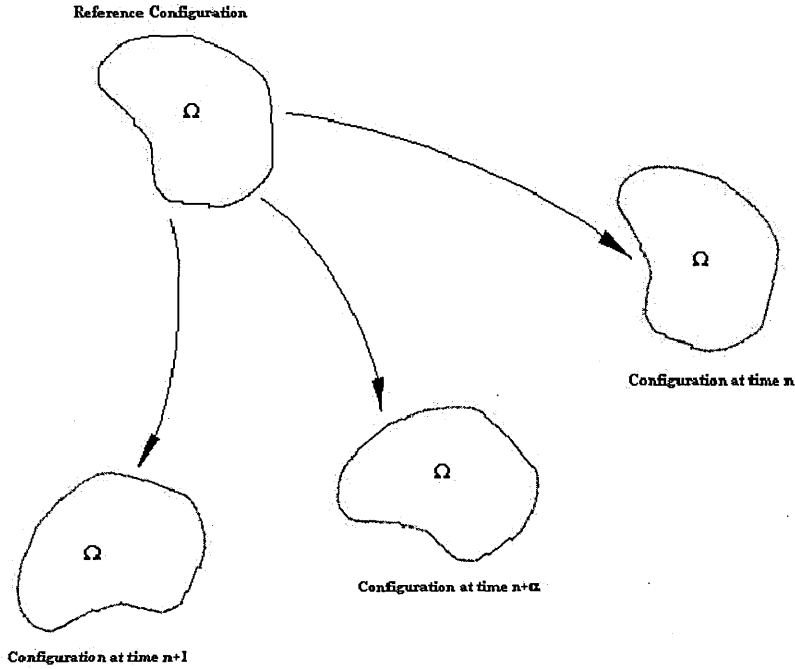


Figure 3.2: Reference configuration advocated by Simo and Tarnow [ 81 ].

Most of the fully explicit codes do not satisfy this condition because contact constraints are enforced with respect to the current configuration regardless of the mutual dependence of configuration and contact force [ 10 ], [ 12 ], [ 39 ], [ 40 ], [ 42 ].

The reason of such an approximation is that, using explicit time integration schemes, the contact could be treated as being quasi-static since the difference between the configurations at the beginning and at the end of the time step is so small that the two configurations will be approximately the same. In fact, to be stable these schemes need a time step so small that the configuration is not supposed to undergo large deformations. Hence, it looks reasonable to evaluate the contact forces with respect to the current configuration, which is known. Unfortunately, this violates the Kuhn-Tucker conditions. And, for this reason, the most recent Explicit Finite Element codes evaluate contact forces at the same time of the stress, i.e.  $t = t_{n+1/2}$ .

Finally, the penalty method itself is not free from criticisms. As stated above, when using a penalty method to enforce contact, the time step becomes smaller due to its dependency on value of the penalty parameter. When used with an explicit time integration scheme, which does not provide in itself numerical damping the solution exhibits typical high frequency oscillations. Laursen and Chawla [ 56 ] demonstrated that when the penalty method is used with an implicit time integration scheme to analyse a fully conservative system the response unconditionally dissipative (which is physically incorrect).

In this light, the need for hybrid methods which treat the contact combining the best aspects of implicit and explicit schemes has been proposed [ 21 ], [ 91 ].

### 3.1.3 Contact in Meshless Methods

The contact boundary condition has been largely ignored in the conventional Smooth Particle Method (SPH). Contact between two bodies is simply handled through the conservation equations, with no restriction on particles from one body being treated as neighbour particles of a particle in another body. Interpolating over neighbour particles regardless of their material type is not only physically incorrect but also results in penetration and mixing that occurs at the contact interface, as SPH does not require the velocity field to be single valued. Typically the mixing occurs within the distance of two smoothing lengths around the interface. Monaghan proposed a modification to the SPH method to prevent the interpenetration, which he called XSPH. While Monaghan's modification does prevent interpenetration it does nothing to solve two other problems with treating contact through the conservation equations:

- ▶ Generation of non-physical tensile forces, resisting separation of two bodies.
- ▶ Generation of shear stresses preventing friction-less or low friction sliding.

A number of contact algorithms have been proposed for treatment of contact between particles and an FE mesh [ 4 ], [ 46 ]. Attaway et al. use a master slave algorithm and enforce the particle – FE mesh contact constraint with the penalty method. Their method could be extended to work with SPH to SPH contact. The drawback would be that one would have to define a master surface to calculate the penetration of the slave nodes through the master segments. This means the construction of a surface on the boundary nodes/particles would be required. Attaway et al.'s contact treatment will be described in more detail in the chapter on SPH-FE coupling (see section 4.1.1).

Chen and Wang [ 24 ] use a similar node to surface contact detection method and use the mixed transformation or singular boundary kernel methods (see section 2.1.5) to enforce the contact condition. Again a major drawback is the need for a contact surface to be defined to calculate the penetration. The applications considered in this paper are metal forming (punch stretching and upsetting) where the nodes are in contact with a, assumed rigid, punch or die. This of course simplifies the definition of the master segments as they remain constant throughout the calculation. Gunther et al [ 37 ] also used the mixed transformation method to model fluid-structure interaction.

The paper by Campbell, Vignjevic and Libersky [ 19 ] describes the development and testing of a contact algorithm for SPH. They define the position of the boundary using the approach proposed by Libersky [ 77 ]. After having determined the boundary nodes determined in this way a particle to particle contact algorithm was developed for 2D. This approach removes the need to define the material boundary as a line in 2D or a surface in 3D, and has many similarities with Belytschko's pinball contact algorithm [ 10 ], [ 11 ]. However, due to non-uniqueness of the surface normal at vertices it was necessary to calculate two surface normals for each boundary particle, and detect a corner particle when the angle between the two normals exceeds a specified angle. The penalty formulation was used to enforce the contact condition, and several equations for

the penalty force calculation were considered. The contact algorithm was tested for one and two dimensional problems for the velocity range between 0.2 and 4.0 km/s to determine the best penalty force equation and the best approach for applying the contact force. The examples used to demonstrate the performance of this technique were a flyer plate impact and the impact of two rubber balls.

### 3.2 New Contact Algorithm

#### 3.2.1 Monaghan's Repulsive Force

A number of treatments have been proposed in an attempt to rid SPH of the instability problem, which can occur in material under compression as well as tension and causes the unphysical clumping of particles. In extreme cases, the errors introduced by the instability cause numerical fracture which is often impossible to distinguish from genuine material failure. Recently Monaghan [ 71 ] introduced a 'short range repulsive term' in the form of an artificial stress between neighbour particles, mimicking the inter-atomic forces of a real material. The stress was shown to effectively relieve the problem of particle clustering.

Monaghan's idea potentially provides an elegant solution to the problem of contact in SPH. The work presented in the following paragraphs evaluates a contact algorithm based on Monaghan's repulsive stress.

The new contact algorithm simplifies the calculation of direction of contact forces and avoids the problems related to non-uniqueness of the surface normal at particles. This is achieved by assigning a contact potential to all boundary particles and defining the contact force as the gradient of this field. Including all material boundary particles in contact in the kernel sum enforces the contact constraints.

Monaghan's repulsive force [ 71 ] increases as the separation between the two particles  $i$  and  $j$  decreases. Its direction and magnitude is defined by the kernel function and distribution of particles.

$$f_{ij} = \frac{W(r_{ij})}{W(\Delta p)} \quad (\text{Eq. 3.7})$$

Where  $\Delta p$  denotes the average particle spacing in the neighbourhood of particle  $i$ .  $\Delta p$  is proportional to the interpolation length scale  $h$ , which determines the kernel support size.

The equations of motion for inviscid fluid with the repulsive force included take the following form [ 71 ]:

$$\frac{\partial v_i}{\partial t} = - \sum_j m_j \left( \frac{p_j}{\rho_j^2} + \frac{p_i}{\rho_i^2} + R f_{ij}^n \right) \cdot \frac{\partial W_{ij}}{\partial x_i} \quad (\text{Eq. 3.8})$$

Where  $n > 0$  and the factor  $R$  depends on the pressure  $p$ , density  $\rho$  and a user specified constant  $\epsilon$ :



$$R = \varepsilon \frac{p}{\rho^2}$$

### 3.2.2 Contact Potential and Contact Force

The main problem in defining contact between bodies discretised with particles consists in defining the location of the boundary of the bodies.

In an SPH approximation, the particle position can be regarded as the centre of a sub-domain with the size defined by the smoothing length.

Recalling the definition of gap function:

$$g_n = g(\mathbf{x}^{(j)} - \mathbf{x}^{(i)}) \cdot \mathbf{n} \quad (\text{Eq. 3.9})$$

Where  $\mathbf{x}^{(i)}$  and  $\mathbf{x}^{(j)}$  are position vectors for particles  $I$  and  $J$  respectively.

$$g = g(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = g(\|\mathbf{x}^{(j)} - \mathbf{x}^{(i)}\|)$$

Considering two particles belonging to two different body, say  $\Omega_A$  and  $\Omega_B$ , the gap function can be rewritten as:

$$g(\mathbf{x}) = \|\mathbf{u}^{(A)}(\mathbf{x}) - \mathbf{u}^{(B)}(\mathbf{x})\| - 2h \quad (\text{Eq. 3.10})$$

Where  $\|\cdot\|$  is a norm representing the distance between the two particles and  $h$  is smoothing distance. Notice that  $g$  is a scalar function that does not depend on the of the unit vector normal to the interface.

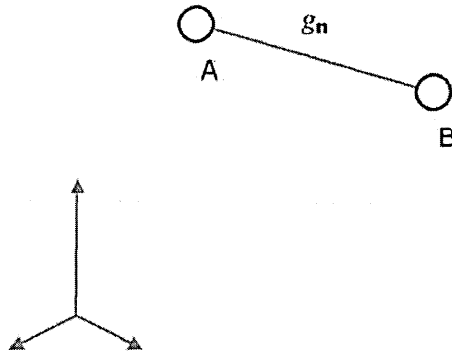


Figure 3.3 Schematic representation of the penetration.

The contact between two bodies occurs when the penetration  $g_n$  becomes  $> 0$ , i.e. the distance between the particles becomes smaller than two times the smoothing length.

Following the idea of body forces being defined as the gradient of a potential field [ 93 ], one could try to use a similar approach to define contact forces. In this contact forces are generated within a boundary layer around the interface rather than purely on the interface.

The contact potential function has to be carefully chosen in order to model contact between to bodies accurately. Intuitively one can see that the potential function should satisfy the following conditions:

- The potential should be zero in the interior of the domain.
- The potential should always be positive.
- The potential should become larger as the distance between to points decreases.

To consider contact, using the proposed approach, a contact force term is simply added to the differential form of the momentum equation.

$$\nabla \cdot \sigma + b + b_c = \rho a \quad \text{in } \Omega \quad (\text{Eq. 3.11})$$

Where  $b_c$  is the contact force. The boundary conditions are defined as:

$$\begin{aligned} \sigma \cdot n &= \bar{t} & \text{on } \Gamma_t \\ \sigma \cdot n &= 0 & \text{on } \Gamma_{to} \\ u &= \bar{u} & \text{on } \Gamma_u \end{aligned} \quad (\text{Eq. 3.12})$$

Where  $\Gamma_t \cup \Gamma_{to} \cup \Gamma_u = \Gamma$ , and  $\Gamma_t \cap \Gamma_{to} = \emptyset$ ,  $\Gamma_{to} \cap \Gamma_u = \emptyset$ ,  $\Gamma_t \cap \Gamma_u = \emptyset$ .

The procedure to obtain the discretised equations is identical to Chapter 2. The first step in the derivation of a weak form of the conservation of momentum equation is to multiply equation Eq. 3.11 by a test function  $\delta w$  and integrate it over the domain of interest  $\Omega$ . This results in:

$$\int_{\Omega} \delta w \nabla \cdot \sigma \, dV + \int_{\Omega} \delta w b \, dV + \int_{\Omega} \delta w b_c \, dV = \int_{\Omega} \delta w \rho a \, dV \quad (\text{Eq. 3.13})$$

This equation can now be transformed to the familiar form (see Chapter 2):

$$\begin{aligned} & \int_{\Omega} \rho [N]^T [N] \, dV \{\ddot{d}\} + \int_{\Omega} [B]^T \{\sigma\} \, dV \{d\} \\ & - \int_{\Omega} [N]^T \{b\} \, dV - \int_{\Omega} [N]^T \{b_c\} \, dV - \int_{\Gamma_t \cup \Gamma_u} ([N]^T \{\sigma\}) \cdot n \, d\Gamma = 0 \end{aligned} \quad (\text{Eq. 3.14})$$

Which, when integrated yields:

$$[K]\{d\} + [M]\{\ddot{d}\} - \{f_b\} - \{f_c\} + \{f_e\} = 0$$

or,

$$[M]\{\ddot{d}\} + \{f_i\} - \{f_b\} - \{f_c\} + \{f_e\} = 0, \quad (\text{Eq. 3.15})$$

Where contact force is defined as:

$$\{f_c\} = \int_{\Omega} [N]^T \{b_c\} dV \quad (\text{Eq. 3.16})$$

This far, the nature of the force  $b_c$  has not been discussed in detail. Having in mind the requirements of the contact potential it is clear that the repulsive force potential used by Monaghan to prevent tensile instability could be a good starting point.

Defining the contact potential as:

$$\phi(x_i) = \int_{\Omega_c} K \left( \frac{W(x_i - x)}{W(\Delta p_{avg})} \right)^n dV \quad (\text{Eq. 3.17})$$

Where  $\Omega_c$  is defined by the intersection between the kernel support centred at a particle in contact with the domain of the body in contact that it does not belong to. It is clear that by defining the contact potential as above all the intuitive conditions on the contact potential are met:

- The potential is zero in the interior of the domain.
- The potential is always be positive.
- The potential becomes larger as the distance between to points decreases.

This integral in Eq. 3.17 can be approximated by point-wise integration:

$$\phi(x_i) = \sum_j^{N_{CONT}} \frac{m_j}{\rho_j} K \left( \frac{W(x_i - x_j)}{W(\Delta p_{avg})} \right)^n \quad (\text{Eq. 3.18})$$

The point-wise integration preserves the completely meshless character of the method. A different numerical integration would require the definition of a boundary surface, which as mentioned earlier is not practical.

Using this contact potential the contact force can be defined as:

$$b_c(x_i) = \nabla \phi(x_i) = \sum_j^{N_{CONT}} \frac{m_j}{\rho_j} K n \frac{W(x_i - x_j)^{n-1}}{W(\Delta p_{avg})^n} \nabla_{x_i} W(x_i - x_j) \quad (\text{Eq. 3.19})$$

And finally the contact force vector can be evaluated by substituting Eq. 3.19 into Eq. 3.16:

$$f_c(x_i) = \int [N_i]^T \{b_c(x)\} dV = \int [N_i]^T \left\{ \sum_1^{NCONT} \frac{m_l}{\rho_l} K_n \frac{W(x - x_l)^{n-1}}{W(\Delta p_{avg})^n} \nabla_x W(x - x_l) \right\} dV$$

Which using point-wise integration becomes:

$$\{f_c(x_i)\} = \sum_j \frac{m_j}{\rho_j} [N_i]^T \{b_c(x_j)\} = \sum_j \frac{m_j}{\rho_j} [N_i]^T \left\{ \sum_1^{NCONT} \frac{m_l}{\rho_l} K_n \frac{W(x_j - x_l)^{n-1}}{W(\Delta p_{avg})^n} \nabla_{x_j} W(x_j - x_l) \right\}$$

or the integral can be completely linearised:

$$\begin{aligned} f_c(x_i) &= \int [N_i]^T \{b_c(x)\} dV = \{b_c(x_i)\} \int \frac{m_i}{\rho_i} W(x_i - x) dV = \{b_c(x_i)\} \frac{m_i}{\rho_i} \\ &= \sum_j^{NCONT} \frac{m_j}{\rho_j} \frac{m_i}{\rho_i} K_n \frac{W(x_i - x_j)^{n-1}}{W(\Delta p_{avg})^n} \nabla_{x_i} W(x_i - x_j) \end{aligned}$$

It is important to notice that direction of the contact force is determined by the SPH approximation of the gradient of the contact potential.

The momentum equation with the contact force included has the following form:

$$\frac{\partial v_i}{\partial t} = \left[ \sum_j m_j \left( \frac{\sigma_j}{\rho_j} + \frac{\sigma_i}{\rho_i} \right) \nabla W_{ij} \right] - \frac{f_{ci}}{m_i} \quad (\text{Eq. 3.20})$$

The contact force is applied to boundary particles as soon as they get within  $2h$  from each other where  $h$  is the smoothing distance. The approach is in keeping with the meshless techniques in general and its implementation in 3D is not complex. The contact algorithm allows surfaces to come together and separate in a physically correct manner.

### 3.2.3 Variational Consistency of the Proposed Contact Algorithm

With respect to the variational consistency it can be pointed out that for the proposed contact algorithm and the expression for the contact force potential, the contact force itself is derived using Galerkin method approach. No assumptions were made on the contact potential function, but that it depends on the distance between the two bodies. Therefore, the only condition for the existence of the contact force potential is that the potential function has to be smooth enough. The contact constraint is imposed directly through the shape functions of the particles in contact, which control the penalty stiffness at the same time.

### 3.3 Numerical Examples

In order to test the performance of the contact algorithm a number of contact problems were analysed. The following analyses are described in detail: 1D wave propagation test, 1D bar impact, 2D plate impact, Water sloshing in 2D, 3D hypervelocity impact.

#### 3.3.1 1D Wave Propagation

Two versions of this test are used. The first one is simply applying a fixed velocity to one side of a bar. The result is a stress wave (step-function) that travels through the bar. In the second version of the test the prescribed velocity is only applied for a limited period of time. This results in a stress pulse (block wave) travelling through the bar. These tests are used in [ 80 ] to test tied coupling between FE and SPH, however a master-slave contact algorithm is also tested.

The problem setup of this test consists of a steel bar of 1 meter long. The left side of the bar is given a prescribed velocity of 0.001 cm/ $\mu$ s. The material model is a linear elastic model with the material constants as in Table 3.1. The bar is discretised in 1D with 100 particles and the smoothing length is equal to the particle spacing. In the case where the prescribed velocity is only applied for a limited time this time was chosen to be 35 $\mu$ s.

Material Constant	Value
Young's Modulus E	210 GPa
Poisson's Ratio $\nu$	0.28
Density $\rho$	7.83 g/cm <sup>3</sup>

Table 3.1: Elastic Material Constants for Wave Propagation Test

The bar was discretised as two pieces of 50 cm with the contact algorithm between the two parts. This test problem immediately exposed a weakness in the contact algorithm. The two sections of the bar are initially in contact. The particle displacement will be small because the only movement at the contact is due to the wave propagation. In the contact algorithm as it was presented the particles start interacting as soon as they are in each others neighbourhood (ie. less than twice the smoothing length away from each other). But because the bars are initially in contact and the smoothing length is equal to the particle spacing the particles at the contact surface are already in each others neighbourhood. This means that as soon as the analysis starts they would be subject to a contact force trying to push the particles away from each other. Due to this contact force stress waves would be generated that travel away from the contact point. Reducing the smoothing lengths at the contact would not provide a solution as the particles would then hardly interact when the stress wave (due to the prescribed velocity of one end of the bar) reaches the contact point. The reason for this is that the contact force is based on a kernel function and hence is very small for particles far away from each other. The material displacements would be quite small as the particle displacements are due to the stress wave. Another solution could be to set the contact force to zero when the pressure is negative or zero. However this also does not provide a satisfactory solution because due to numerical round-off small positive pressures may exist (these are pressures several orders of magnitude below the pressures in the stress wave). But this is enough to switch on the contact. Because the particles are already

halfway in each others kernel supports the force applied is big and continues to be applied until the particles are outside each others kernel supports.

An effective solution to this problem is very simple. It consists of setting the contact force to zero for particles that are further than the particle spacing away from each other. This means that they will be included in the contact search when they are closer than twice the smoothing length away from each other, but they only see a contact force when they are less than the particle spacing away. The force does not increase gradually from zero as would be the case if the smoothing length was reduced for the contact, but immediately a large force.

With this change made to the contact algorithm the results of the wave propagation test look very similar to the master-slave algorithm tested by Sauer [ 80 ]. The results also show the reflected wave observed by Sauer when using a standard master-slave algorithm. The reflected wave is slightly less pronounced with the contact algorithm based on a contact potential.

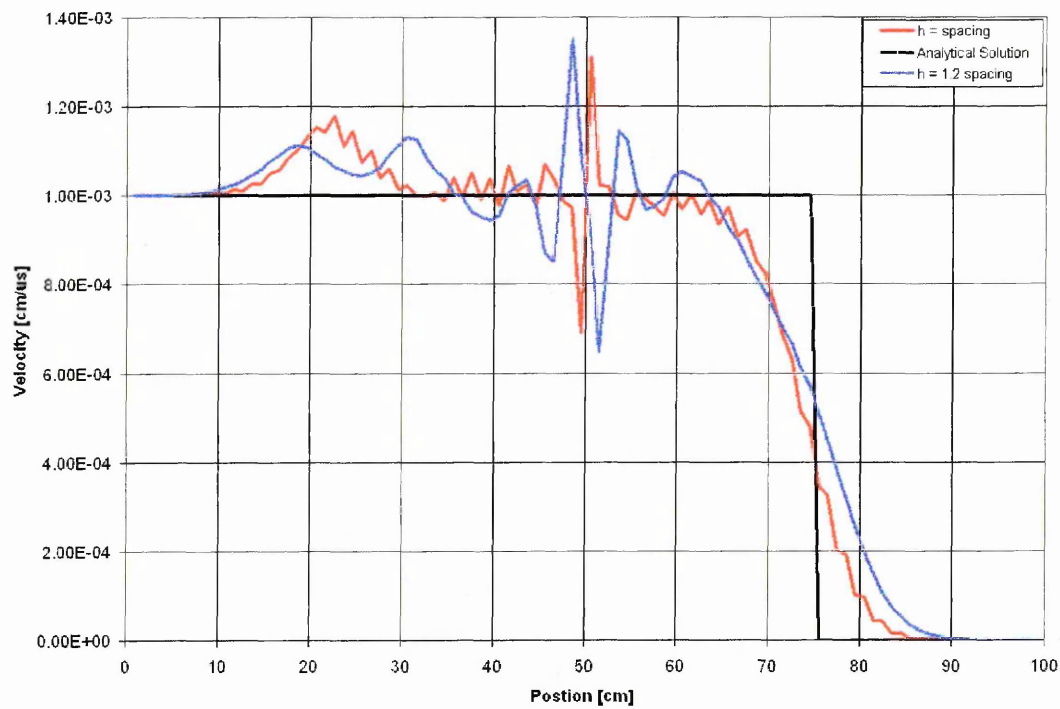


Figure 3.4: Velocity Profile for Wave Propagation Problem, at time = 128.8μs.

Using a smoothing length equal to 1.2 times the particle spacing results in slightly smoother results. However in this case the oscillations at the contact surface are more pronounced.

The second test is identical in setup to the first test, but the prescribed velocity is set to 0 after 35μs. The results are shown in Figure 3.5. Again the results are similar to the ones reported in [ 80 ].

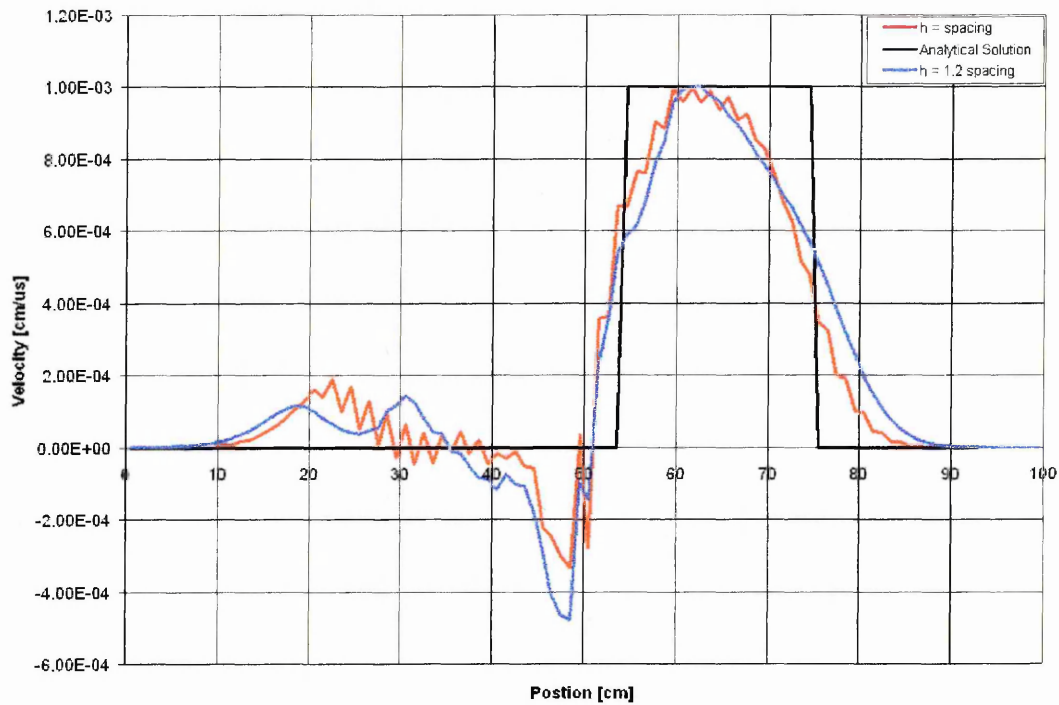


Figure 3.5: Velocity Profile for Block Wave Propagation Problem, at time = 128.8  $\mu$ s.

These results show a similar accuracy compared to a master-slave algorithm. However the main purpose of this contact algorithm is to use it to simulate water impact, mainly characterised by large displacements and deformations. The tests in this paragraph are wave propagation problems characterised by small displacements and small deformations.

### 3.3.2 1D Bar Impact

The performance of the new contact algorithm can be further tested by simulating a 1D elastic bar impact. The aim of this was to further test the effectiveness of the contact force algorithm, in exclusion from any effect that could occur in 2D from the calculation of the vector along which the contact force was applied.

The test problem used was a symmetrical impact of two 0.4 cm long steel bars, see Figure 3.6. 100 particles were placed along the 1D line, 50 particles in each bar. The smoothing length was 0.01 cm and the initial inter-particle space was equal to the smoothing length. The space between the two contact particles was 0.01, so there was no initial penetration. Each bar was given an equal but opposite initial velocity of 200 m/s.

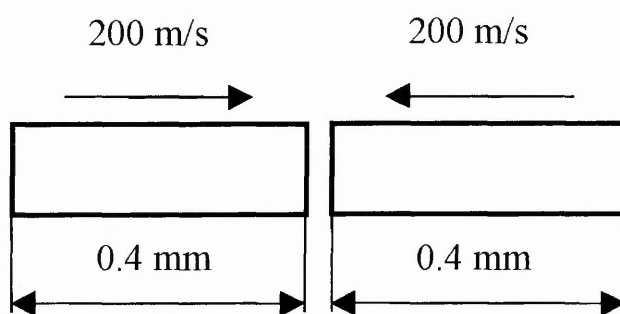


Figure 3.6: Schematic representation of the 1D test problem

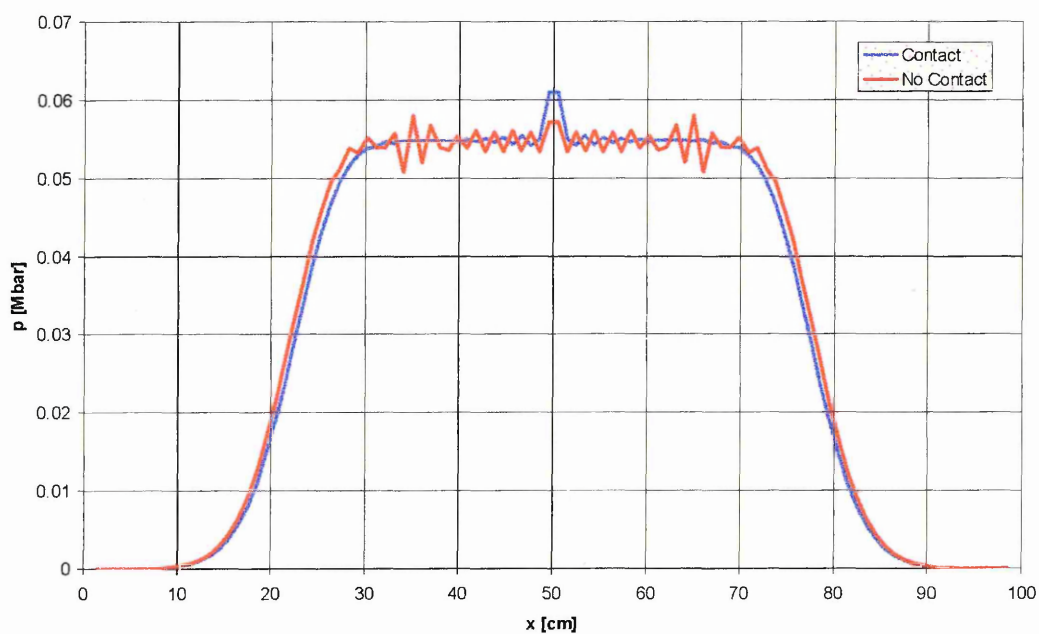


Figure 3.7: Pressure profiles with and without contact at  $48 \mu\text{s}$ .



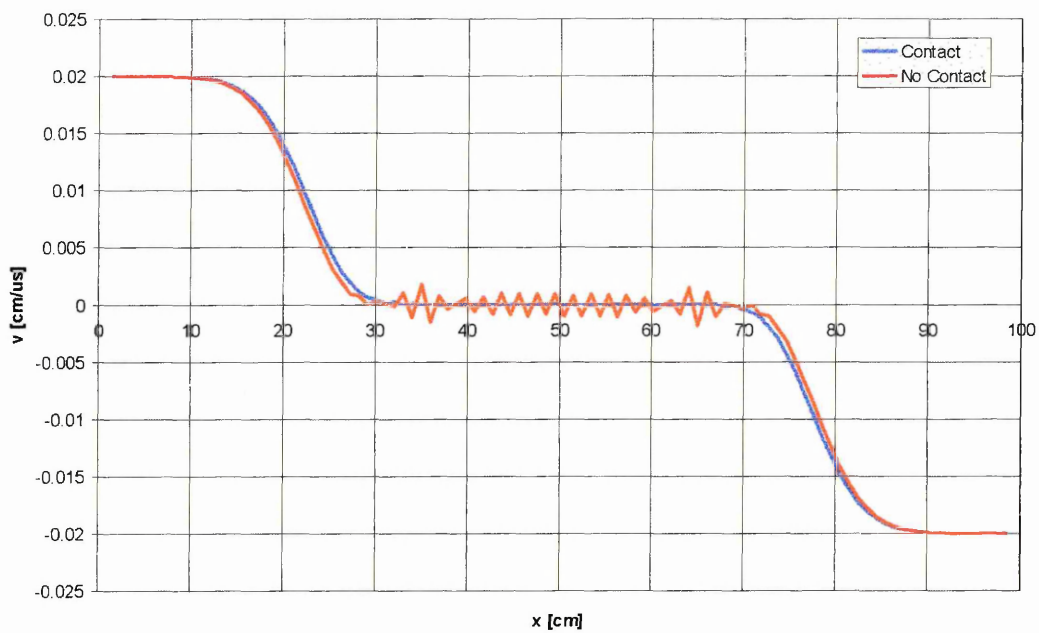


Figure 3.8: Velocity profiles with and without contact at 48  $\mu\text{s}$ .

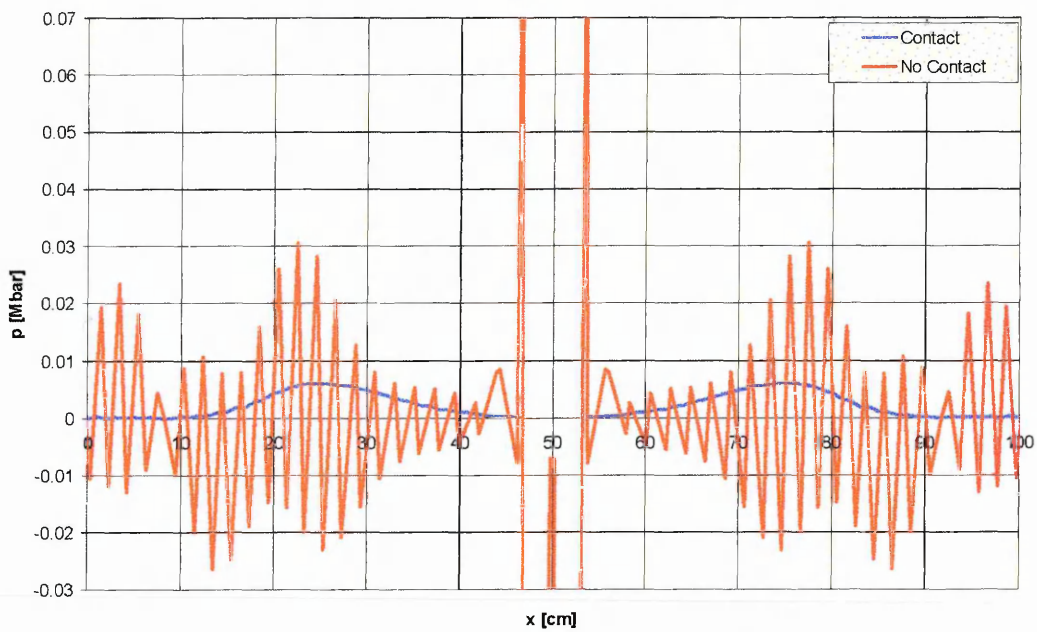


Figure 3.9: Pressure profiles with and without contact at 300.0  $\mu\text{s}$ .

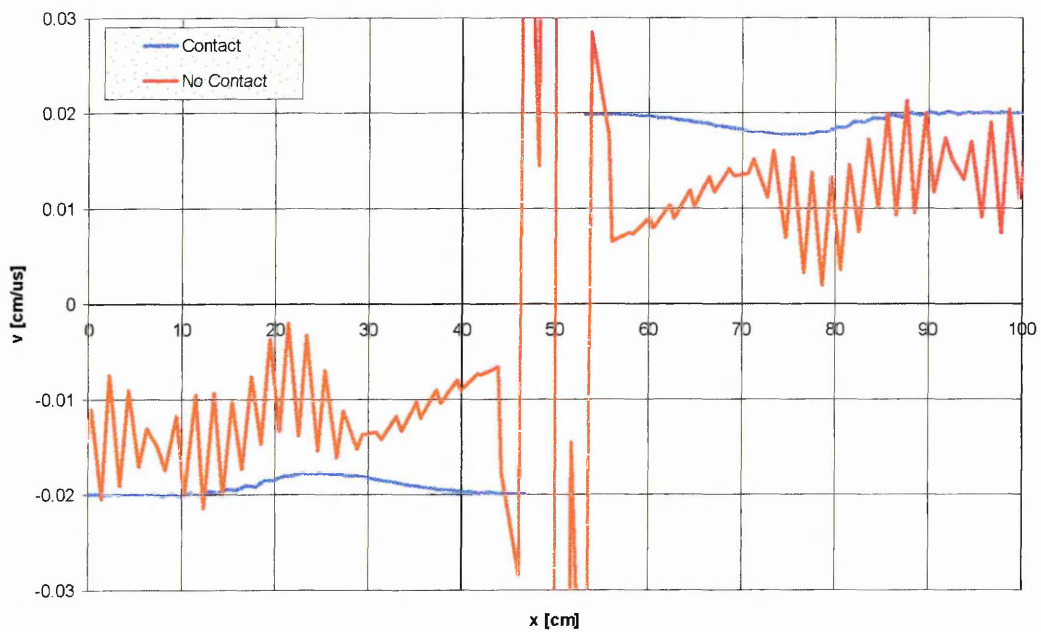


Figure 3.10: Velocity profiles with and without contact at 300.0  $\mu$ s.

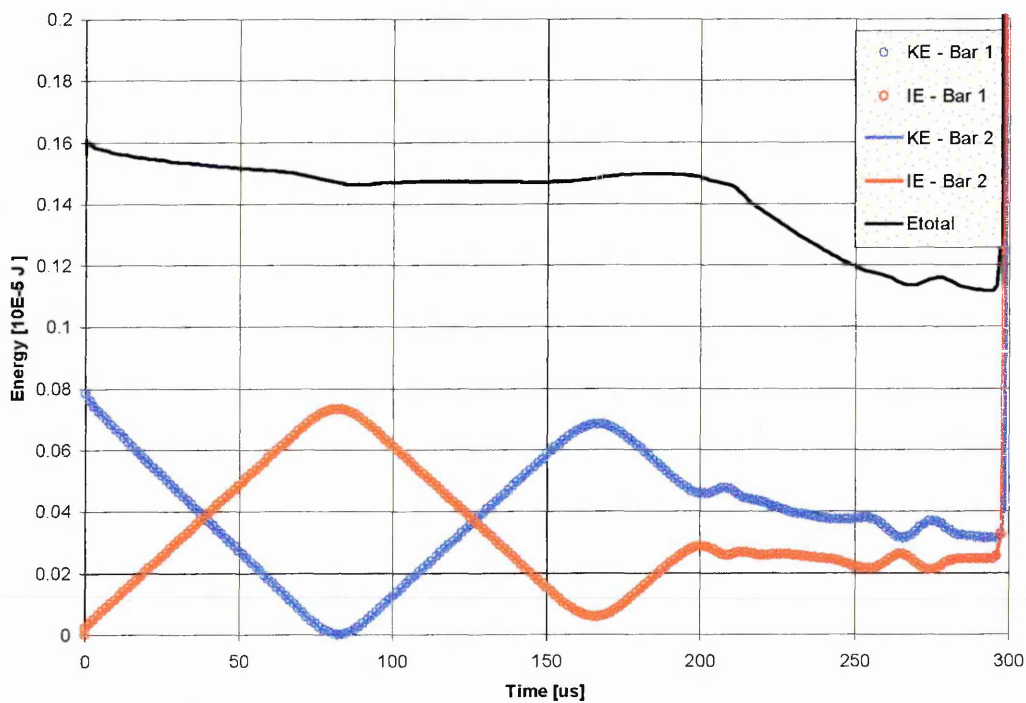


Figure 3.11: Kinetic - and Internal Energy Time Histories without contact.

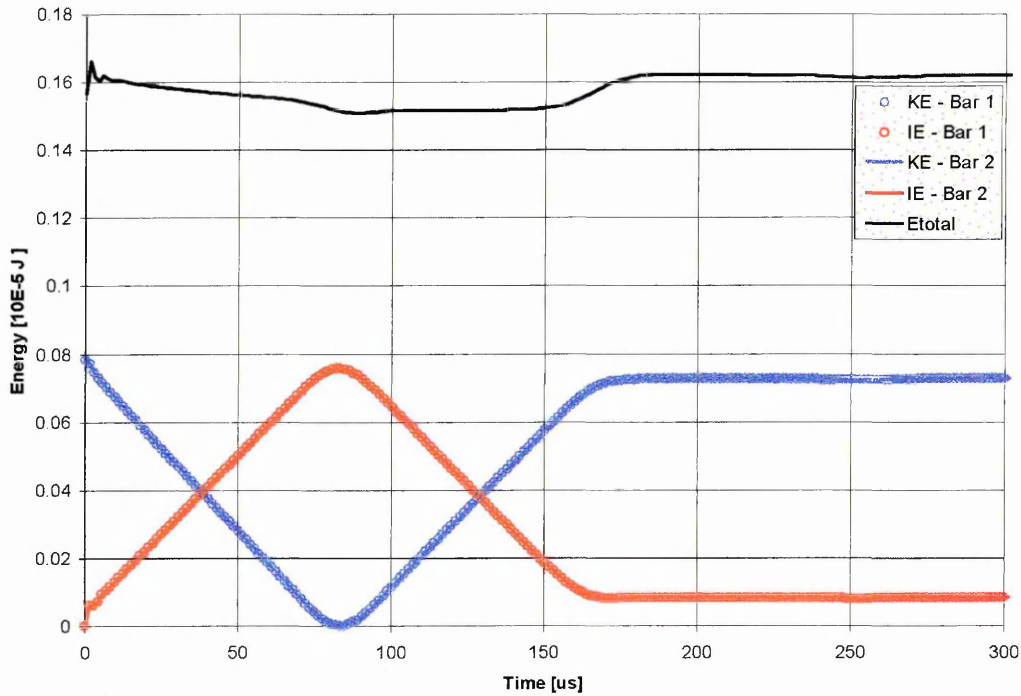


Figure 3.12: Kinetic - and Internal Energy Time Histories with contact.

To provide a reference point the problem was initially run without contact, using kernel sums to allow interaction between the two bodies. With the kernel sums the two end particles were initially in contact. Figure 3.7 and Figure 3.8 shows the resulting pressure and velocity profiles at time  $48.8\mu\text{s}$ . The shock waves are propagating away from the point of contact. At this point no significant differences can be seen between the two sets of results. Figure 3.9 and Figure 3.10 shows the pressure and velocity profiles at time  $300.0\mu\text{s}$ , with the profiles for kernel contact shown for comparison. At this point the release waves have just reached the point of contact. One can see that with the contact algorithm, apart from a small elastic wave, the bars are unloaded. No non-physical tensile stresses are generated. This is in contrast with the use of kernel contact where non-physical tensile stresses and tensile instability around the point of contact develop. From the velocity plot it can be seen that when the contact algorithm is used the bars are travelling away from each other with a speed almost equal and opposite to the initial impact speed. Considerable oscillation can be seen in the stress and velocity at the interface for kernel contact. This oscillation was described by Swegle [ 83 ] who observed it when modelling the 1D impact of two initially separated bodies using kernel contact. This oscillation corresponds to a zero energy mode, for which stress field does not cause particle accelerations that would reduce the stress amplitude. Swegle showed that this was a characteristic of the SPH method, as the SPH estimate to the stress field has its local maxima and minima at the particles giving the stress field gradient of zero. In the SPH method no acceleration will be produced at a particle if the stress at the two neighbour particles is equal, this is independent of the stress at the particle itself. This is the stress version of the SPH zero-energy mode where an alternating velocity field does not generate stresses in the material to resist the

particle motion. Figure 3.11 and Figure 3.12 shows the exchange between kinetic - and strain energy as a function of time.

### 3.3.3 2D Plate Impact

As the 1D tests showed that the contact based on the contact potential performs satisfactorily, its performance was tested in 2D. The contact algorithm was tested in 2D by modelling the normal symmetrical impact of two steel blocks, and a fluid sloshing simulation. The block impact test allows for the evaluation of the capability of the contact algorithm in dealing with corner contact, where at one side the material is in contact, and the other side is a stress-free surface. The fluid sloshing problem evaluates the ability to deal with frictionless sliding at the contact interface, and extreme deformations and change in contact 'surface'.

For the symmetrical block impact each block was 1 cm by 0.4 cm and consisted of 50 particles by 20 particles, giving a total of 2000 particles in the whole model. The two blocks have an initial relative velocity of 400 m/s, see Figure 3.13. The contact boundary particles in each body were initially spaced  $h$  apart, so the initial penetration was zero. The materials were modelled with an elastic-plastic material model, with material properties that are typical for steel.

The simulation results for this problem are shown in Figure 3.14 to Figure 3.16. Again the same simulation was performed without contact. These results are shown (on the LHS) for comparison. In Figure 3.14 one can see the pressure in the blocks after  $1\mu\text{s}$ . The block is at this point in compression, the release waves have not yet reached the centre of the block. From these results it is clear that the contact algorithm has no problems in dealing with contact on corners. In Figure 3.15 the pressure in the blocks is shown just after the point where the release waves have reached the contact line (after  $2\mu\text{s}$ ). One can see the typical tensile stresses generated when no contact algorithm is used, while the results on the RHS show the beginning of the separation between the two unloaded blocks. Figure 3.16 shows the complete separation of the blocks at  $5\mu\text{s}$ .

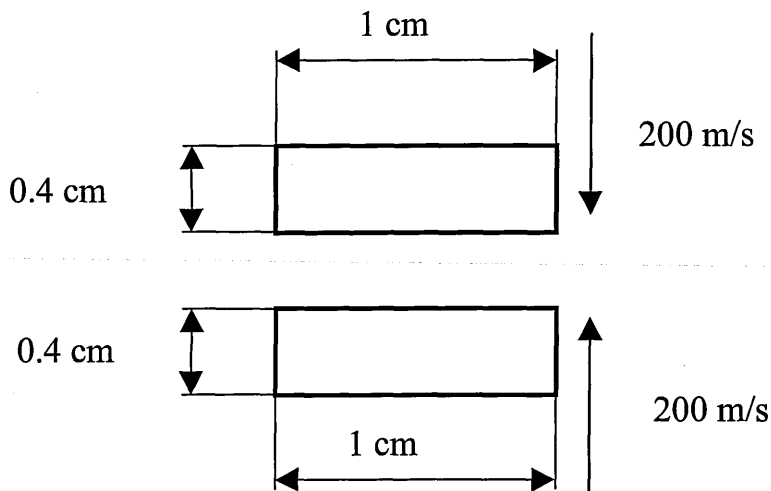


Figure 3.13: Symmetric Block Impact in 2D: Problem Setup

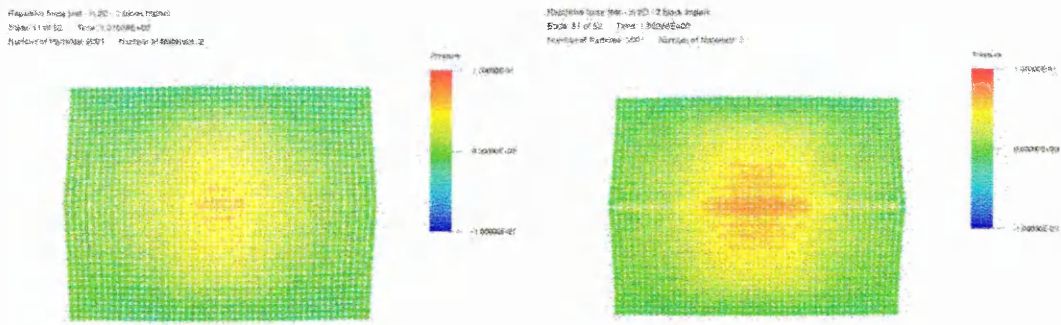


Figure 3.14: Pressure plots 1 $\mu$ s after impact, without contact (LHS) and with contact (RHS)

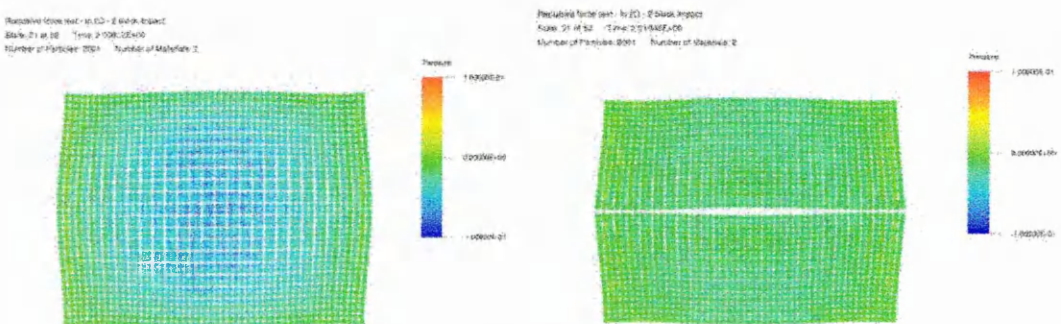


Figure 3.15: Pressure plots 2 $\mu$ s after impact, without contact (LHS) and with contact (RHS)

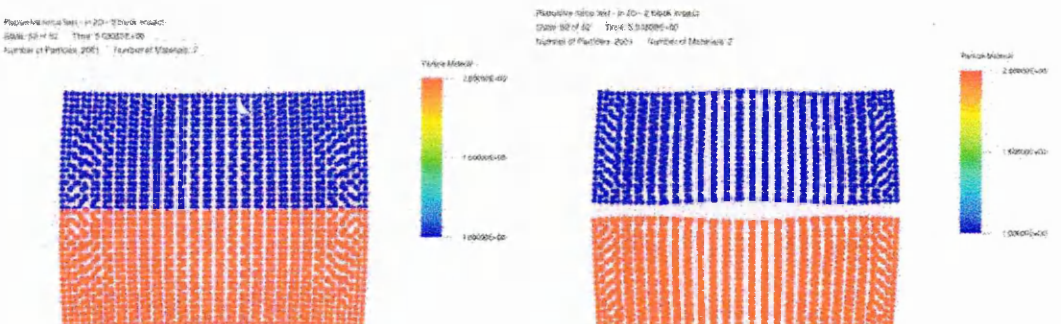


Figure 3.16: Materials plots 5 $\mu$ s after impact, without contact (LHS) and with contact (RHS)

### 3.3.4 Fluid Sloshing in 2D

The fluid sloshing simulation is the second 2D test that was performed. The dimensions of the container are shown in Figure 3.17. The fluid is modelled as a viscous fluid with the following equation of state to ensure that the material behaves quasi-incompressible:



$$p = B \left[ \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right], \tag{Eq. 3.21}$$

where  $B$  and  $\gamma$  are user defined constants. The volume of fluid is initially at rest and moves under the influence of a gravity field. The response time considered was 3s. Figure 3.18 shows the fluid flow as it impacts the wall of the container, and the subsequent oscillatory settling of the fluid due to gravity.

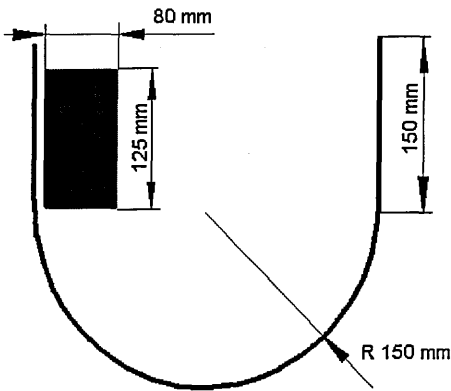


Figure 3.17: Fluid Sloshing Simulation: Problem Setup.

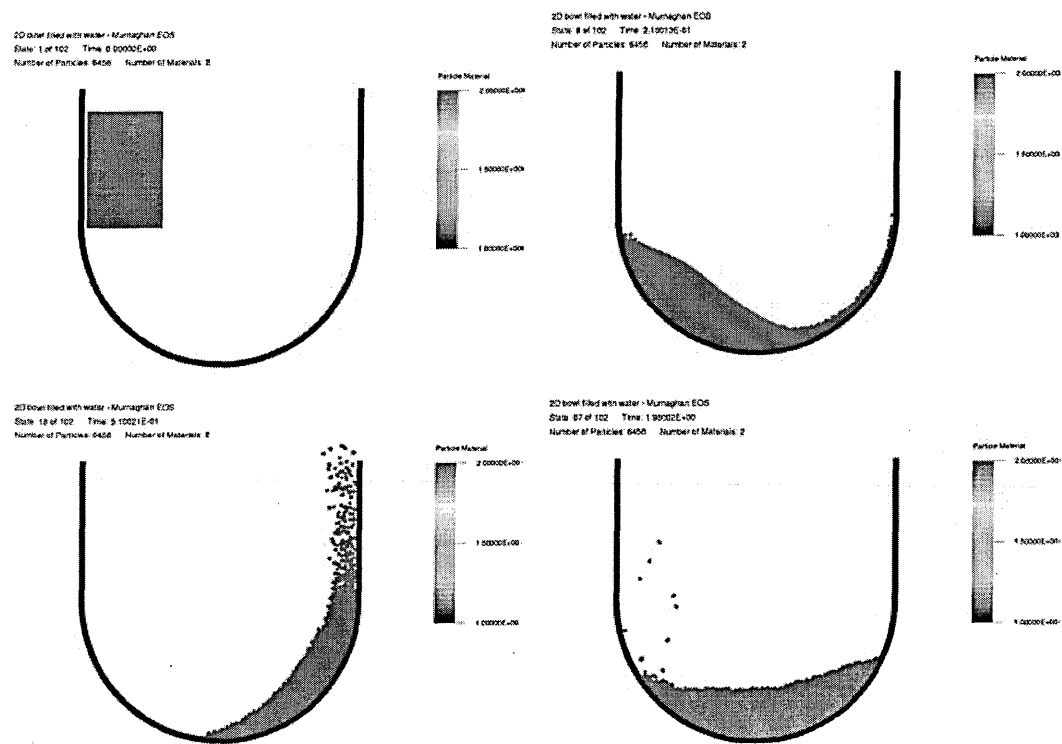


Figure 3.18: Position of the fluid at 0.00, 0.21, 0.51 and 1.98 s.

### 3.3.5 3D Hypervelocity Impact

In order to demonstrate that this contact algorithm performs in 3D a hypervelocity impact of a sphere on a plate was carried out. This type of problem is a severe test for the robustness of the contact algorithm as both objects are subjected to extremely large deformations. The radius of the sphere is 0.3mm, the plate thickness and radius is 0.2mm and 0.15mm. The material of both objects is aluminium and an elastic-plastic hydrodynamic material model with Gruneisen equation of state was used. The sphere has an initial velocity of 7km/s. Due to the symmetry of the problem a quarter model was created (see Figure 3.19). Cross section plots after 0.0, 0.1, 0.2 and 1.5  $\mu$ s are shown in Figure 3.20 and Figure 3.21 which shows that the contact algorithm is performing adequately. No experimental data, such as hole diameter or debris cloud angle, was available to perform a more qualitative validation of the simulation results.

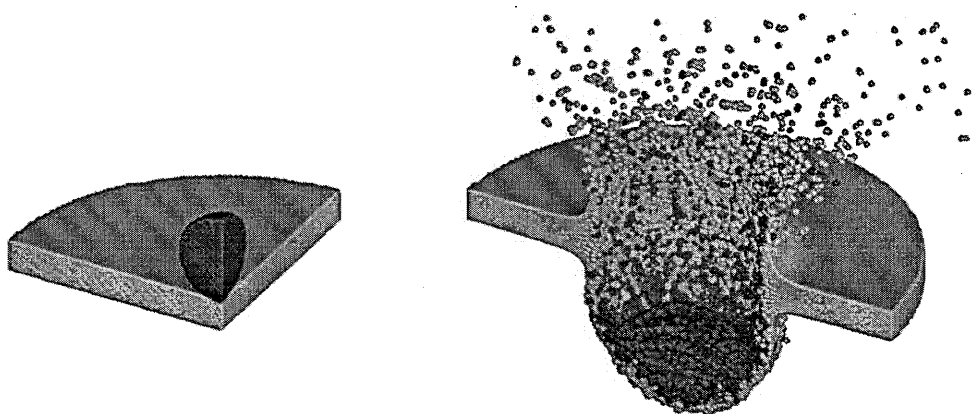
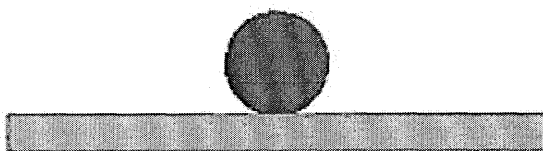


Figure 3.19: Isometric View of 3D Hypervelocity Impact at 0.0 and 0.2 $\mu$ s.



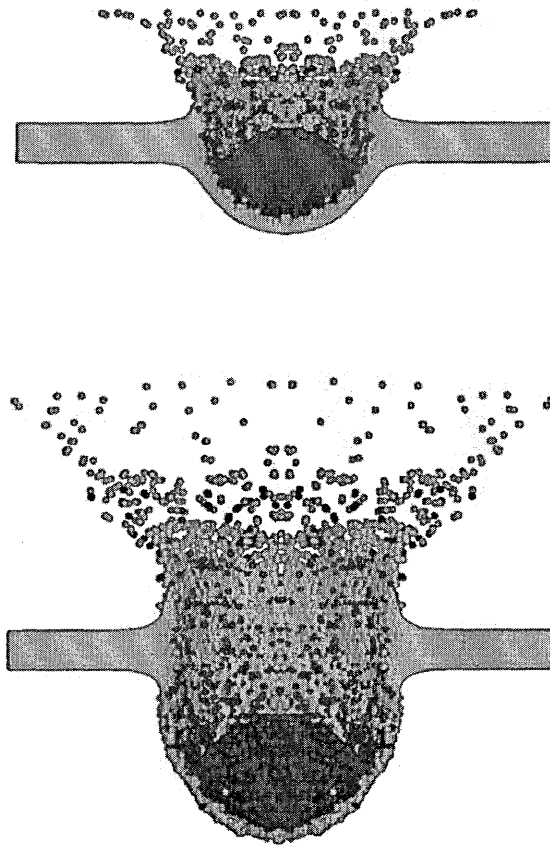


Figure 3.20: Cross Section Plots of 3D Hypervelocity Impact at 0.0, 0.1 and 0.2 $\mu$ s.

### 3.4 Conclusions

A conceptually simple and computationally efficient method for modelling contact in SPH with the following characteristics has been proposed.

The detection of contact and calculation of contact force are based on quantities that have to be calculated within an SPH algorithm.

There is no need to construct contact surfaces, to explicitly define normals to the surface and to calculate interpenetration.

The new contact algorithm was successfully applied to 1D, 2D and problems.

When used with the conventional SPH method it is equivalent to particle to particle contact.

The proposed approach can be used with the collocated and non-collocated normalised SPH method.



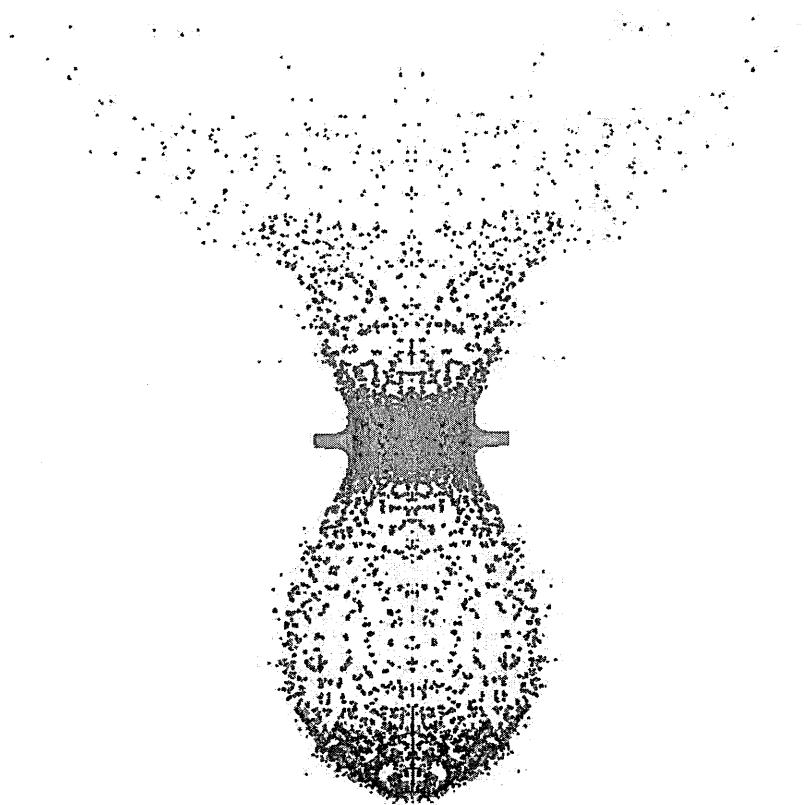


Figure 3.21: Cross Section Plot of 3D Hypervelocity Impact at 1.5 $\mu$ s.

## Chapter 4 Coupling of SPH Solver with DYNA3D

### 4.1 Introduction

The objective of this chapter is to present a method of coupling an SPH solver to LLNL-DYNA3D. The advantage of being able to use a combination of SPH particles and FE is that it should allow for improved simulation results when modelling for example the interaction of water with a deforming and collapsing structure. The detailed modelling of the complex fluid structure interaction during an impact of an aircraft on water or soft soil are such cases that could benefit from a coupled FE-SPH modelling capability. Other problems that would benefit from a combined FE-SPH modelling capability are ballistic impact or penetration problems. In the remainder of the introduction an overview of the existing methods of coupling of SPH and FE will be given.

#### 4.1.1 Previous work on coupling of SPH and FE

A number of papers and reports have been published on coupled FE-SPH solvers. The earliest attempts were by Attaway, Heinstein and Swegle [ 4 ], and Johnson [ 46 ]. Recently a very complete and rigorous study of FE-SPH coupling was performed by Sauer [ 80 ]. The common denominator in these authors' work is that their algorithms are all based on a master-slave algorithm to model contact.

Attaway, Heinstein and Swegle [ 4 ] use the master-slave algorithm to couple an SPH solver to PRONTO. PRONTO is an FE hydrocode developed at Sandia National Laboratories. The algorithm for interaction between elements and particles is described for a 2D case and is based on a predictor-corrector method. First they test whether a particle (slave) could penetrate the finite element surface (master) in the current timestep. First the accelerations of the FE nodes and the SPH particles are calculated assuming that no contact occurs:

$$\text{FE } \hat{a}_i = \frac{F_i}{m_i} \quad (\text{Eq. 4.1})$$

$$\text{SPH } \hat{a}_i = \sum_j m_j \frac{\sigma_j + \sigma_i}{\rho_j \rho_i} \nabla W \quad (\text{Eq. 4.2})$$

Using these accelerations, the velocities and displacements are calculated. These new nodal positions are used to perform the contact check. If there is contact between an SPH particle and an FE segment then this constraint is enforced by applying a contact force to both the slave node and master surface.

The contact force on the slave particle is calculated such that the interpenetration  $\delta$  is removed during the next timestep:

$$F_s = \frac{m_s \delta}{\Delta t^2} \quad (\text{Eq. 4.3})$$

The application of the contact force in this manner results in an iterative method. The reason for this is that it has been assumed that the master surface will not deform due to the contact conditions. To take into account the deformation of the master surface one has to correct the contact forces. The contact forces on the master nodes are calculated by interpolation using intrinsic coordinates, and then converted into nodal accelerations. Due to this acceleration on the master surface the penetration will be smaller than initially calculated. Therefore Attaway et al. apply a corrector step as follows: The acceleration of the contact point  $a_{cp}$  on the master surface is calculated from the nodal accelerations using the interpolation with intrinsic coordinates. From this acceleration a corrected reaction force  $F_{cs}$  can be calculated as:

$$F_{cs} = \frac{m_s \delta}{\Delta t^2} - a_{cp} m_s. \quad (\text{Eq. 4.4})$$

Using these new reaction forces the new master accelerations can be calculated. This iterative process could be repeated, but the authors report that one iteration is usually sufficient.

Johnson [ 46 ] uses a master slave algorithm where, when a slave particle penetrates the master segment by a certain pre-specified amount, the following three conditions are enforced:

- ▶ Conservation of linear momentum.
- ▶ Conservation of angular momentum.
- ▶ Normal velocity components of the particle and the master segment are identical.

Finally Sauer [ 80 ] also uses a master slave algorithm to compare his adaptive FE-SPH algorithm to. The algorithm uses a 'gap'. This means that the particle, when it moves towards the master segment, will experience a contact force when the gap becomes smaller than some predefined value. The contact value for the gap is chosen as one half the particle distance.

## 4.2 Structure of DYNA3D

The purpose of this section is to provide an introduction to the structure of the DYNA3D program. The version of the code described in this section is the 1995 release of public DYNA3D. Public DYNA3D is developed and maintained by the Methods Development Group at Lawrence Livermore National Laboratory. A good overview of the DYNA3D code is provided in [ 17 ]. The other documentation available for DYNA3D is the user manual [ 61 ], and the theoretical manual [ 41 ]. In order to modify DYNA3D it is necessary to have an understanding of the structure of the code. In this section the emphasis will be on describing these parts of the code that are relevant to the coupling of the code with a meshless solver. The DYNA3D code was

first developed in the mid-70's. The code can handle 3D geometry only, and it is an explicit finite element code which makes it suitable for the analysis of problems involving large deformations and inelastic materials. The finite element library contains the following linear elements: eight node bricks, four node shells and two node beams. Higher order elements are computationally more expensive and for this reason not available. It contains several contact-impact algorithms allowing different bodies to interact, along with material models and equations of state that cover a wide range of material behaviour. DYNA3D is written in Fortran77.

#### 4.2.1 Basic Structure of DYNA3D

The code can be split into five main sections: Input, Initialise, Restart, Solution and Output. The relationship between these sections is shown in Figure 4.1. Depending on whether a new analysis is started, or an old analysis continued different section of the program are called. The input and initialisation sections are called when a new analysis is started. These sections read in the text input file, and initialise the contact algorithm logic, calculate the lumped mass matrix and calculate any other values required for the specific option used in the analysis. All data read in from the input file, and data generated in the initialisation section is stored in the main database of the program. This database is essentially a vector which contains a sequence of data blocks, such as nodal positions and velocities, element connectivity, material data, element field variables etc.

The restart section is used if a problem is being restarted from a dump file. This section first reads in a short text file which supplies the new termination time. Then the dump file is read, which contains a copy of all the solution variables and data structures at the solution time that it was written.

The solution section is the main part of the program and performs the explicit time integration. The output section does not have a single main subroutine. Specific subroutines are called from the solution section at specified time intervals. DYNA produces three main output files, dump files, state plot files that contain all results for a specified solution time, and time history files.

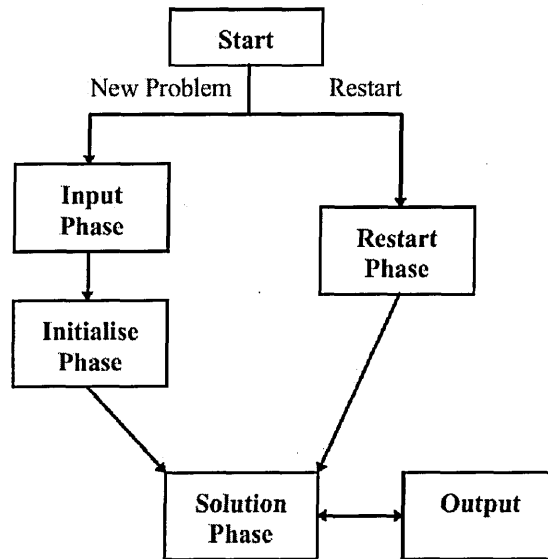


Figure 4.1: Basic Structure of DYNA3D.

#### 4.2.2 Data handling

DYNA3D is designed to be as efficient as possible in its use of memory, this was important at the time when the code was developed as the amount of memory available at the time was very limited. At the core of DYNA is the central database which, as mentioned earlier, consists of a single large array. In this database all the solution variables are stored. This consists of data for example the nodal coordinates and velocities, element connectivity, stress tensor and contact surface segments. In order to access the data DYNA3D calculates a number of pointers. These pointers are integers which refer to the location in the database of each variable or array stored there. The pointers are calculated during the input phase.

Variables are passed to subroutines by using these pointers in the call statements to pack and unpack the data from the main database into these sub-routines. By these packing and unpacking operations the data can be transformed into arrays of different shapes. The following example can clarify this. Consider for example the nodal positions. In the database all data is stored as a string of numbers. So if the database is an array  $a(*)$ , then a pointer  $lc11$  points at the position where the nodal coordinates are stored. If a subroutine needs the nodal positions then they can be made available by passing them as an argument in the call statement:

```
call move(a(lc11))
```

in the definition of the subroutine one can then use the following lines to make the nodal positions available as a 3 x number of nodes sized array:

```
subroutine move(x)

  real : x(3,*)
```

This system of packing and unpacking can make it hard to trace variables in the program, especially since the variables names, or even their data structure, is not always the same in all subroutines.

The other way by which variables are passed between subroutines is by the use of common blocks. Sometimes the names in the common blocks are different from subroutine to subroutine. This can also make it hard to trace variables through the program.

#### 4.2.3 Solution Section

The solution section forms the main section in DYNA3D. This subroutine contains the loop that performs the time integration and contains many calls to other subroutines. Depending on the options that are used in the specific analysis DYNA3D will call different subroutines in the main solution loop. In this section only the main subroutine which is common to all analyses will be looked at.

The basic structure of the main solution loop subroutine is shown in Figure 4.2. At the centre of the time integration loop are the following equations:

$$\mathbf{a}^n = \mathbf{M}^{-1}\mathbf{F}^n \quad (\text{Eq. 4.5})$$

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \mathbf{a}\Delta t^n \quad (\text{Eq. 4.6})$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1/2}\Delta t^{n+1/2} \quad (\text{Eq. 4.7})$$

Between the calculation of the accelerations, and the velocities and displacements the dynamic relaxation, if active, is carried out. The state plot, time history and optional output files are written. Next, the solution time is checked, if it has reached the end time the run is terminated, and a dump file is written. After this check the point where a new or restarted analysis starts is reached.

Following this point the acceleration/force array is zeroed, and the calculation of the new force vector is started. The contributions from surface tractions, concentrated loads, body force loads and the effect of non-reflecting boundary segments, are calculated. Then the elements are processed. A different subroutine is called for each basic type of element: solid elements, beam elements, shell elements, thick shell elements and discrete elements. Each routine calculates the contributions of the element type under consideration to the internal force vector:

$$\sum_n \int_{V_n} \mathbf{B}^t \sigma dV \quad (\text{Eq. 4.8})$$

where  $n$  is the number of elements of that type. This is the element stress contribution to the weak form of the equilibrium equation. The steps that are performed in order to

calculate this force array are summarised in Figure 4.3 for the case of hexahedron elements (bricks).

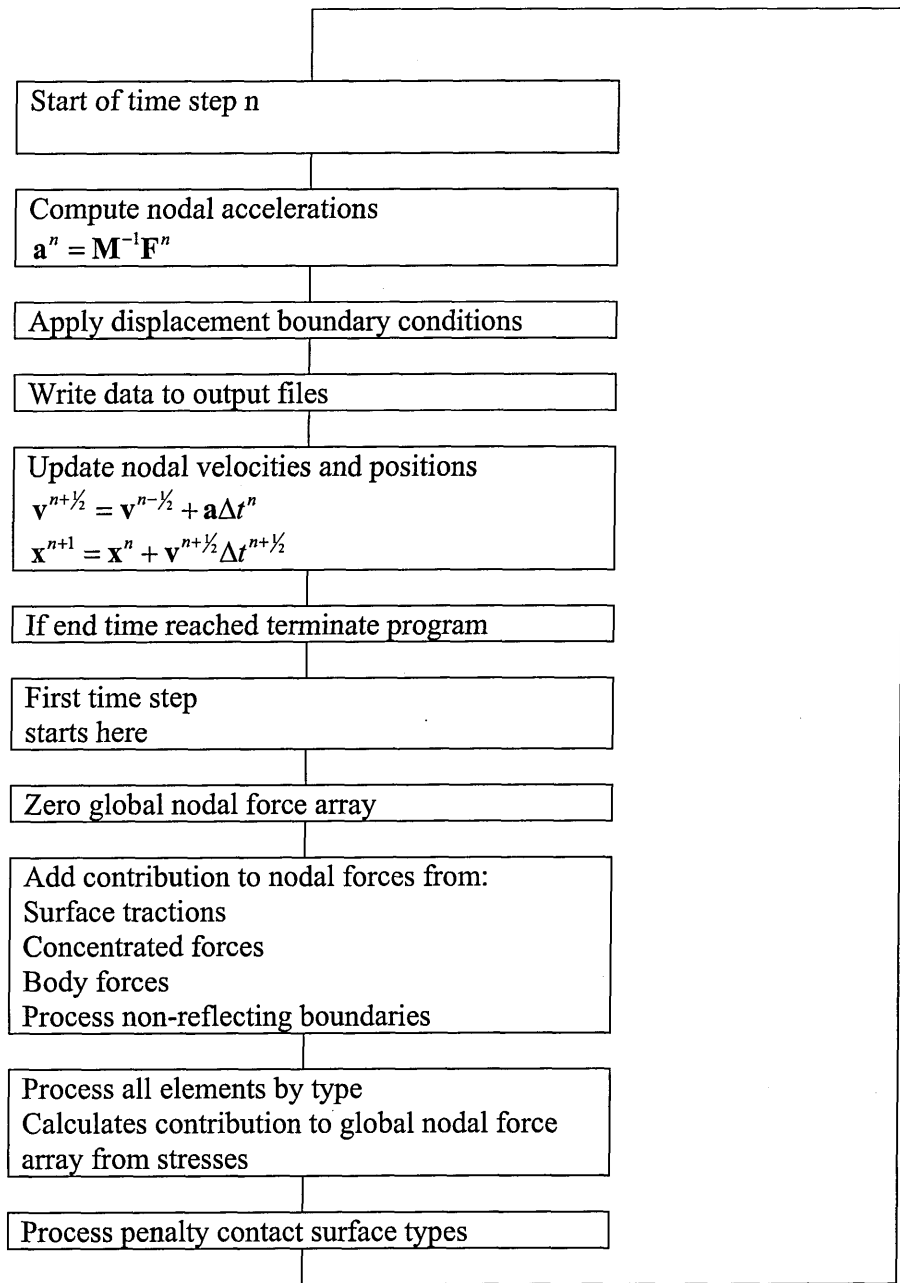


Figure 4.2: Structure of main solution subroutine.

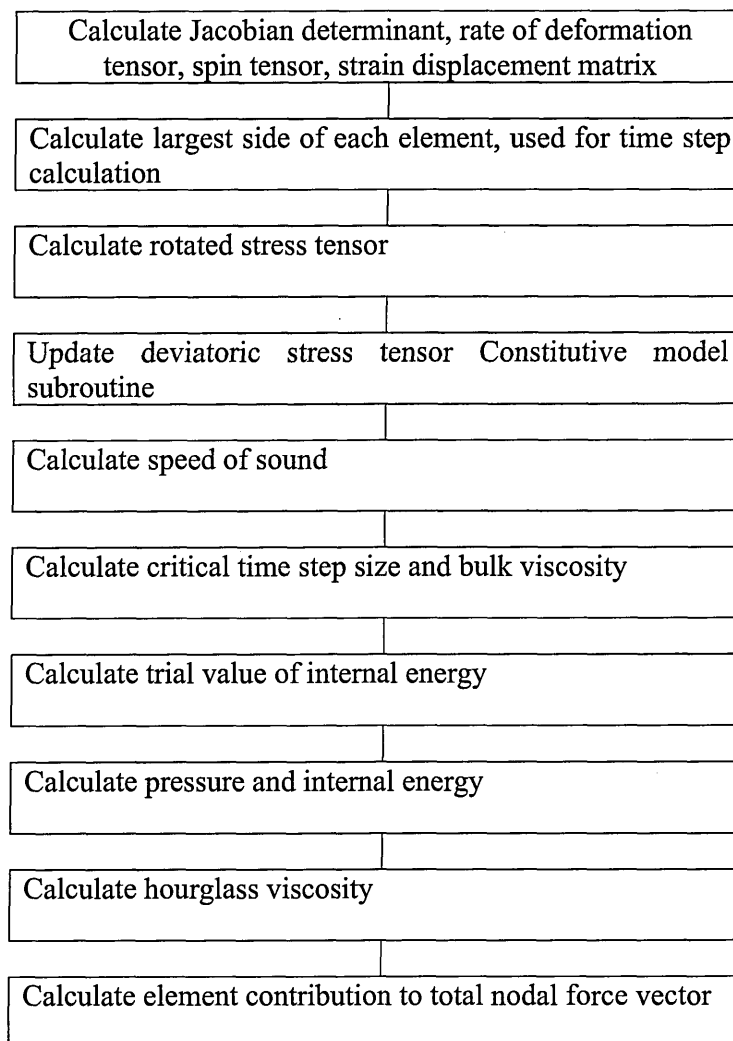


Figure 4.3: Main steps in the calculation of the force vector of hexahedron elements, for strength model requiring an equation of state.

Finally the sliding interfaces that use the penalty stiffness method are processed. This type of contact surface applies a nodal force to enforce the contact constraint. Following this the routine loops back to the start of the loop.

### 4.3 Structure of SPH solver

#### 4.3.1 Basic Structure of the SPH solver

The basic structure of the SPH solver is the same the DYNA3D code (see Figure 4.1 and Figure 4.4), as described in the previous section. This facilitates the merging of the two codes greatly. Similarly to DYNA3D, the SPH solver reads in all necessary data, such as analysis options, nodal information and material data, from a text file. After this input section all variables are initialised. The principal tasks of the initialisation section is to calculate the nodal masses and to calculate any other values required for the



specific options used in the analysis such as initial internal energy and pressures, any history variables, load curves etc. There is no restart option in the current version of the SPH solver. After the initialisation section the main solution loop is entered.

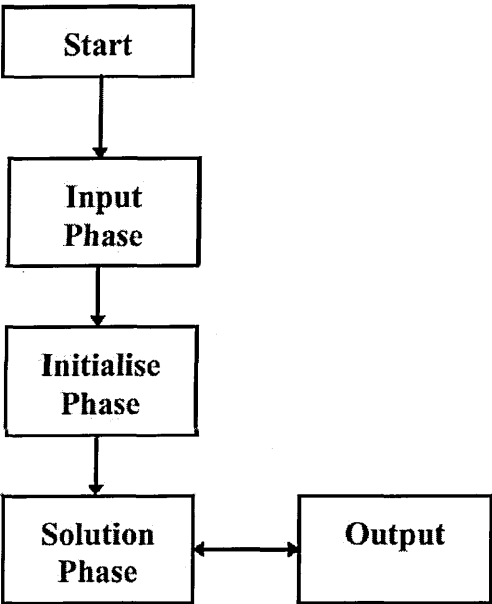


Figure 4.4: Basic Structure of SPH solver.

4.3.2 Data Handling

The SPH solver has been written in FORTRAN90. This allowed for making use of some features that are available in FORTRAN90, but not in FORTRAN77. This is mainly visible in the data structure and data handling of the SPH solver. The data structure of the SPH solver is completely different from the DYNA3D data structure, global variables are defined in MODULES. The effect of this is most evident in the way data is made available for use in subroutines. Local variables can be passed to subroutines in the call statements, but global variables are accessed via USE statements. The SPH solver uses dynamic memory allocation for most variables. This means that exactly the right amount of memory is allocated for every analysis, and that there is no software restriction on the size of problem that can be analysed. The maximum problem size is only limited by the hardware. The memory allocation is done after the input section of the code, and allocates the memory for data such as the nodal coordinates, velocities, accelerations, stress tensor, etc.

### 4.3.3 Solution Section

The solution section is the main part of the program. This subroutine contains the loop that performs the time integration and contains many calls to other subroutines. The basic structure of the subroutine is shown in Figure 4.5. The core of the time integration loop are the following equations:

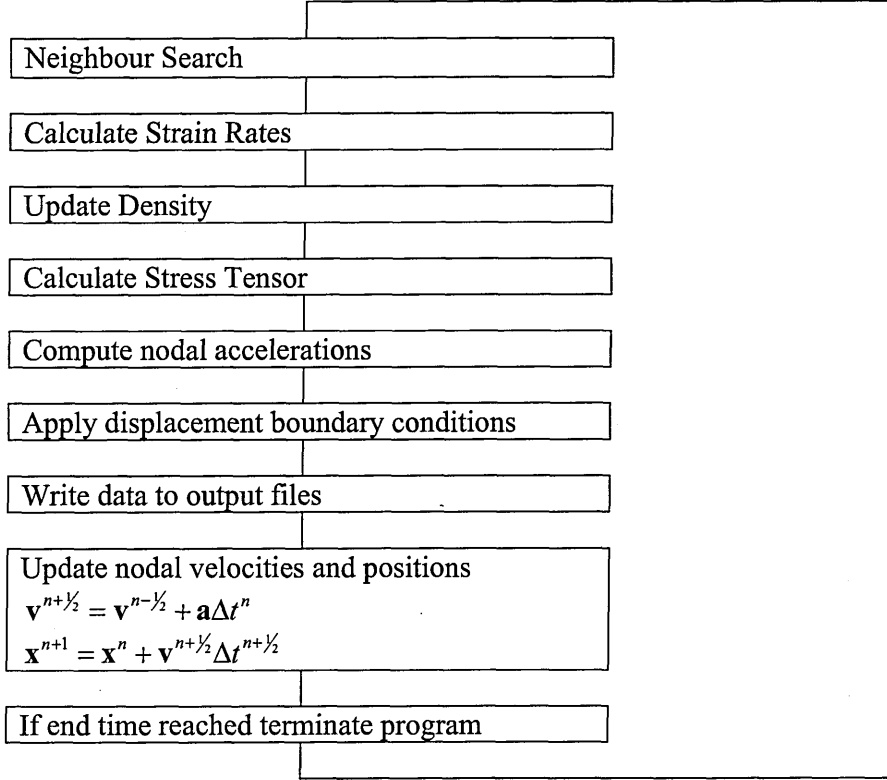


Figure 4.5: Structure of main solution loop of SPH solver.

$$\dot{\epsilon}_i = -\frac{1}{2} \sum_j \frac{m_j}{\rho_j} (v_j - v_i) \frac{\partial W_{ij}}{\partial x_i} - \frac{1}{2} \sum_j \frac{m_j}{\rho_j} (v_j - v_i) \frac{\partial W_{ij}}{\partial x_i} \quad (\text{Eq. 4.9})$$

$$a_i = \sum_j m_j \left( \frac{\sigma_j - q_j}{\rho_j^2} + \frac{\sigma_i - q_i}{\rho_i^2} \right) \frac{\partial W_{ij}}{\partial x} \quad (\text{Eq. 4.10})$$

$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \mathbf{a} \Delta t^n \quad (\text{Eq. 4.11})$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1/2} \Delta t^{n+1/2} \quad (\text{Eq. 4.12})$$

At the beginning of every timestep a new list of neighbour particles is built. This list is built using the linked list algorithm. Once this list is available the strain rates are calculated using Eq. 4.9 followed by an update of the density and the calculation of the stress tensor using an appropriate constitutive model. Once the stress tensor is known

Eq. 4.10 is used to calculate the new accelerations. These steps correspond to the evaluation of Eq. 4.8 and Eq. 4.5 in DYNA3D, and are summarised in Figure 4.6. Between the calculation of the accelerations, and the velocities and displacements the displacement boundary conditions are applied and the state plot, time history and optional output files are written.

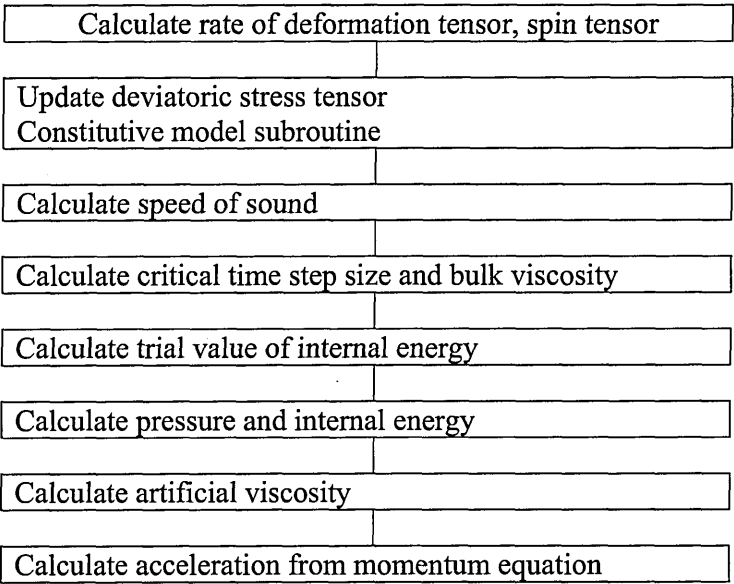


Figure 4.6: Main steps in the calculation of the acceleration vector for SPH particles, for a strength model requiring an equation of state.

#### 4.4 Linking of SPH and DYNA3D codes

##### 4.4.1 Basic Structure of combined DYNA3D – SPH code

The main steps in the coupling of DYNA3D with an SPH code that need to be performed are:

- ▶ The merging of the codes into one executable.
- ▶ The use of the same integration loop and time step.
- ▶ Combining the input and output of both codes.
- ▶ The interaction of SPH particles with the finite elements.

From the previous sections it is clear that the main structure of DYNA3D and the SPH solver are similar. They both have an input, initialisation and solution phase. This allows the codes to be merged without changes to the structure of either code. This main structure has been merged as shown in figure 7. Some of the variable and subroutine names are identical in the two codes. Therefore it was decided to rename all subroutines and variables of the SPH code by adding `mcm_` as a prefix. This will allow for SPH and DYNA3D variables to be used in the same routines.

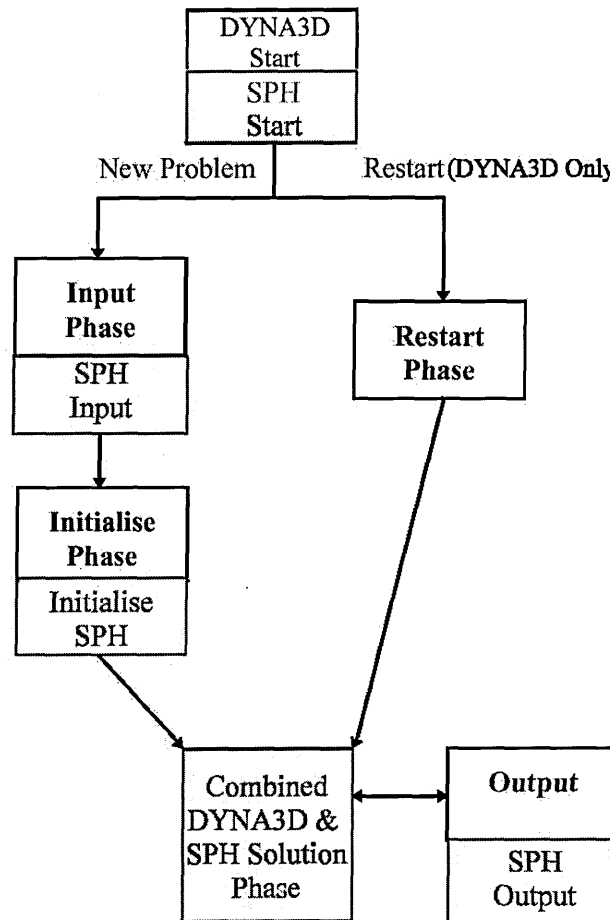


Figure 4.7: Basic Structure of Coupled DYNA3D and SPH code.

The combination of startup, input and initialisation routines is fairly straight forward as they can be performed sequentially. The merging of the main solution loop is more complex. The loop has to be modified such that any operation on particles and elements happens at the same point in the time integration loop. For example, accelerations should be calculated for particles and elements at the same position in the loop, using variables that are known at the same solution time.

#### 4.4.2 Data Handling

Because the data handling is very different for the two codes it would require an almost complete rewrite of one of the codes if the data handling were to be the same for both codes. Therefore two separate data handling structures will be kept in the combined DYNA3D-SPH solver. When DYNA3D and SPH data needs to be known in the same routine the required information can be made available in the SPH code by adding common blocks with DYNA3D data or the data can be passed as arguments in calls to SPH subroutines. If SPH data has to be made available in a DYNA3D routine this will require more effort. The reason for this is that `MODULES` and `USE` statements are a feature only available in Fortran 90. Therefore it is not possible to simply add a `USE` statement in the DYNA3D routine. Two solutions to this are the conversion of the

specific DYNA3D routine into Fortran 90, or alternatively to write the SPH data into a dedicated SPH-DYNA3D interaction COMMON block, which can then be added to the original Fortran 77 DYNA3D routine.

#### 4.4.3 Solution Section

The main solution loops have been combined in such a way that any operation on particles and elements happens at the same point in the loop. This is schematically shown in Figure 4.8. The SPH subroutines are inserted at the points where the equivalent FE operation is performed. The interaction between SPH particles and elements is performed at the same point where contact interfaces are processed. At this point in the solution loop one could modify a penalty contact interface algorithm from DYNA3D to deal with the particles interaction with the elements. This requires the nodal positions of both elements and SPH particles to be available. The resulting contributions to the force vector would have to be made available in both solvers. In this thesis a different approach will be described. This approach treats the FE nodes as SPH particles. The SPH particle to particle algorithm from Chapter 3 will be used to calculate the forces on FE nodes and SPH particles due to their interaction. This coupling algorithm will be described in detail later in this chapter.

#### 4.4.4 Post Processing

A major problem in the combination of the two solvers that needed to be addressed first was the combination of the output, such as plot files. Currently TAURUS is used to visualise the DYNA3D analysis results, while the SPH data is visualised using the EnSight® visualisation software. EnSight is a visualisation package that is able to read in output from several CFD and FE solvers, for example Abaqus, Fluent, LS-DYNA. On top of this EnSight also has its own input format, with which data from any source can be visualised provided it is written out in this format. It is this feature that enables EnSight to be used as a post-processor for an SPH solver. Furthermore EnSight can display discrete elements and nodes in several ways. One option is to simply plot a dot, which can be coloured according to the value of a field variable at the element under consideration. The second option however is to place a rendered and shaded sphere at the position of the element. This feature is extremely useful when visualising SPH results as it gives a much better visual impression of the shape of the domain. The radius of the sphere can be changed in the GUI. Again this sphere can be coloured according to the value of a field variable at the element under consideration, so the particle can be colour with for example pressure data.

TAURUS is not capable of visualising discrete elements with one node, while EnSight has no interface to read in LLNL-DYNA3D output data. Hence it was necessary to either modify TAURUS such that the discrete elements can be visualised, or an EnSight output option had to be added to the DYNA3D – SPH code. Both would require a major effort as data originating from two different data structures will have to be combined into one output file.

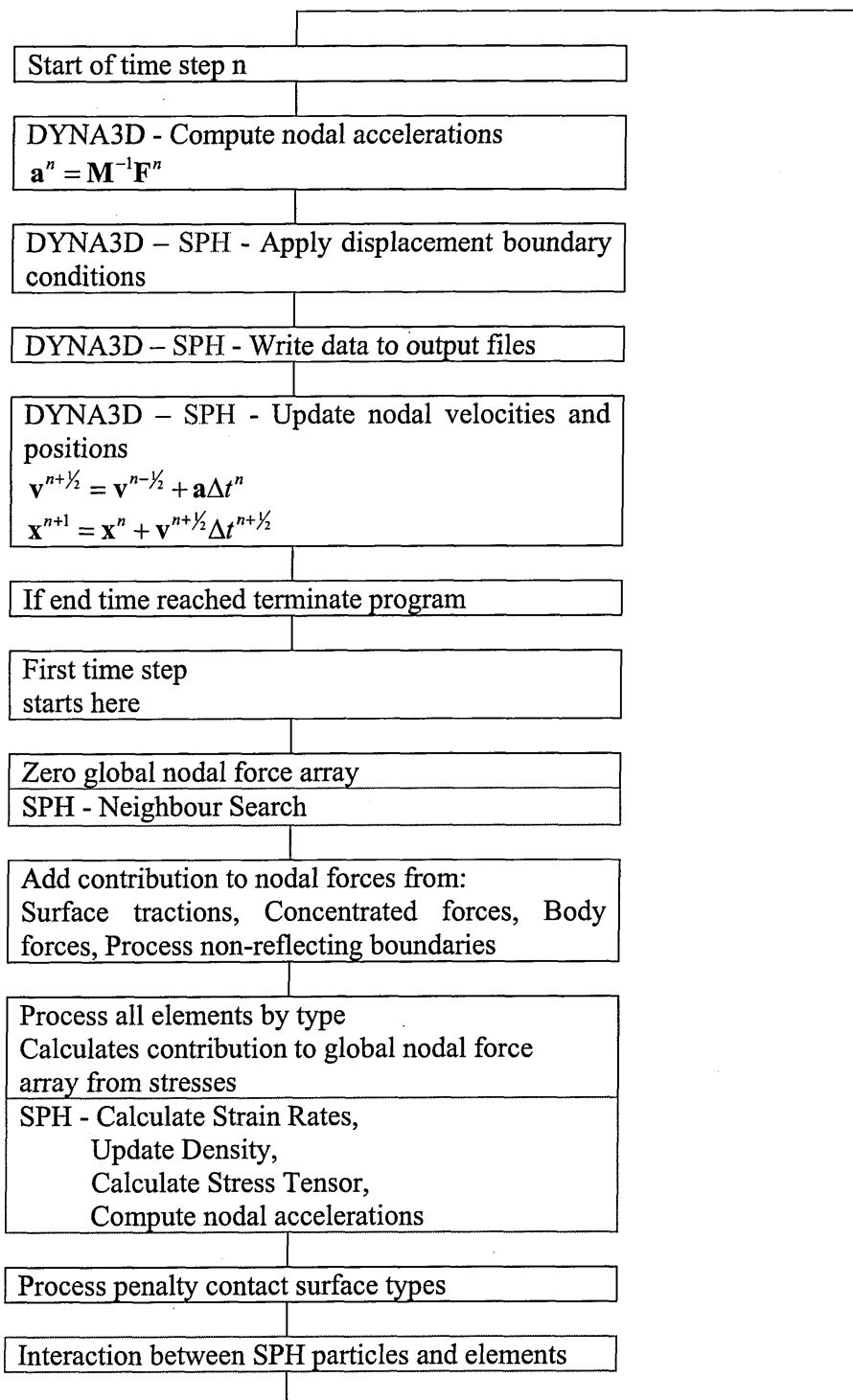


Figure 4.8: Structure of main DYNA3D – SPH solution subroutine.

The modification of TAURUS would require the generation of spheres or cubes in the current visualisation window. The colour of these shapes would have to be made

dependent on any user specified field variable, for example stress components, velocity etc. Furthermore these shapes would have to ‘interact’ with the elements such that a proper hidden view can be plotted. A particle in front of an element will have to be plotted, a particle hidden by elements should not be plotted. These modification would require a thorough understanding of the TAURUS post processor.

Alternatively the DYNA3D information could be written to files in a format readable by EnSight. This would require a complete rewrite of the DYNA3D output section. The EnSight input format consists of a series of ASCII or binary files. The EnSight output routine would have to extract all the data from the main DYNA3D database, and the SPH modules. This data could then be written into the EnSight readable format.

It was decided to opt for the latter method of modifying the DYNA3D output section to write the data in a format readable by EnSight. This required a thorough understanding of the DYNA3D output format, and the way DYNA3D extracts all necessary data from the main database and writes it into a compact binary file. This is described in detail in APPENDIX C.

#### 4.4.5 Contact Potential and Contact Force

To consider contact using the proposed approach, a contact force term is simply added to the differential form of the momentum equation.

$$\nabla \cdot \sigma + b_c = \rho a \quad \text{on } \Omega \quad (\text{Eq. 4.13})$$

Where  $b_c$  is the contact body force, and  $\Omega = \Omega_{FE} \cup \Omega_{SPH}$ . The boundary conditions are defined as:

$$\begin{aligned} \sigma \cdot n &= \bar{t} & \text{on } \Gamma_t \\ \sigma \cdot n &= 0 & \text{on } \Gamma_{to} \\ u &= \bar{u} & \text{on } \Gamma_u \end{aligned} \quad (\text{Eq. 4.14})$$

Where  $\Gamma_t \cup \Gamma_{to} \cup \Gamma_u = \Gamma$ , and  $\Gamma_t \cap \Gamma_{to} = \emptyset$ ,  $\Gamma_{to} \cap \Gamma_u = \emptyset$ ,  $\Gamma_t \cap \Gamma_u = \emptyset$ .

The weak form of these equations is obtained by multiplying equation Eq. 4.13 by a test function  $\delta w$  and integrate it over the domain of interest  $\Omega$ , which results in:

$$\int_{\Omega} \delta w \nabla \cdot \sigma \, dV + \int_{\Omega} \delta w b_c \, dV = \int_{\Omega} \delta w \rho a \, dV \quad (\text{Eq. 4.15})$$

The weak form of this equation is (see Chapter 2) in Voigt notation:

$$\int_{\Omega} \rho \{\delta w\}^t \{\ddot{u}\} \, dV + \int_{\Omega} \{\nabla \delta w\}^t \{\sigma\} \, dV - \int_{\Omega} \{\delta w\}^t \{b_c\} \, dV - \int_{\Gamma_t \cup \Gamma_u} (\{\delta w\}^t \{\sigma\}) \cdot n \, d\Gamma = 0, \quad (\text{Eq. 4.16})$$

and can be discretised in the space domain by using shape functions for the displacement trial functions and the test functions:

$$\begin{aligned} \int_{\Omega} \rho [N]^T [N] dV \{\ddot{d}\} + \int_{\Omega} [B]^T \{\sigma\} dV \{d\} \\ - \int_{\Omega} [N]^T \{b_c\} dV - \int_{\Gamma_i \cup \Gamma_u} ([N]^T \{\sigma\}) \cdot n d\Gamma = 0 \end{aligned} \quad (\text{Eq. 4.17})$$

Which, when integrated yields:

$$[M]\{\ddot{d}\} + \{f_i\} - \{f_c\} + \{f_e\} = 0, \quad (\text{Eq. 4.18})$$

Where the contact force vector is defined as:

$$\{f_c\} = \int_{\Omega} [N]^T \{b_c\} dV \quad (\text{Eq. 4.19})$$

Because the FE and SPH sub-domains only interact via the contact algorithm, the only vector containing coupling terms will be the contact force vector. This means that Eq. 4.18 can be rewritten as two equations, one for the FE subdomain and one for the SPH subdomain:

$$\begin{cases} [M]^{FE} \{\ddot{d}\}^{FE} + \{f_i\}^{FE} - \{f_c\}^{FE} + \{f_e\}^{FE} = 0 \\ [M]^{SPH} \{\ddot{d}\}^{SPH} + \{f_i\}^{SPH} - \{f_c\}^{SPH} + \{f_e\}^{SPH} = 0 \end{cases}$$

Defining the contact potential as in Chapter 3:

$$\phi(x_i) = \int_{\Omega_c} K \left( \frac{W(x_i - x)}{W(\Delta p_{avg})} \right)^n dV \quad (\text{Eq. 4.20})$$

Using this contact potential  $b_c$  can be defined as:

$$\begin{aligned} b_c^{FE}(x_i^{FE}) &= \nabla \phi(x_i^{FE}) = \sum_j^{NSPH} \frac{m_j^{SPH}}{\rho_j^{SPH}} Kn \left( \frac{W(x_i^{FE} - x_j^{SPH})}{W(\Delta p_{avg})} \right)^{n-1} \nabla_{x_i^{FE}} W(x_i^{FE} - x_j^{SPH}) \\ b_c^{SPH}(x_i^{SPH}) &= \nabla \phi(x_i^{SPH}) = \sum_j^{NFE} \frac{m_j^{FE}}{\rho_j^{FE}} Kn \left( \frac{W(x_i^{SPH} - x_j^{FE})}{W(\Delta p_{avg})} \right)^{n-1} \nabla_{x_i^{SPH}} W(x_i^{SPH} - x_j^{FE}) \end{aligned}$$

And finally the contact force vector is:

$$f_c^{FE}(x_i^{FE}) = \sum_j^{NSPH} \frac{m_j^{SPH}}{\rho_j^{SPH}} \frac{m_i^{FE}}{\rho_i^{FE}} Kn \left( \frac{W(x_i^{FE} - x_j^{SPH})}{W(\Delta p_{avg})} \right)^{n-1} \nabla_{x_i^{FE}} W(x_i^{FE} - x_j^{SPH})$$



$$\mathbf{f}_c^{\text{SPH}}(\mathbf{x}_i^{\text{SPH}}) = \sum_j^{\text{NFE}} \frac{m_j^{\text{FE}}}{\rho_j^{\text{FE}}} \frac{m_i^{\text{SPH}}}{\rho_i^{\text{SPH}}} \text{Kn} \left( \frac{W(\mathbf{x}_i^{\text{SPH}} - \mathbf{x}_j^{\text{FE}})}{W(\Delta \mathbf{p}_{\text{avg}})} \right)^{n-1} \nabla_{\mathbf{x}_i^{\text{SPH}}} W(\mathbf{x}_i^{\text{SPH}} - \mathbf{x}_j^{\text{FE}})$$

It suffices to add these force vectors to the contribution of the internal and external forces in the momentum equations of the FE and SPH solvers.

The contact force is applied to boundary particles as soon as they get within  $2h$  from each other where  $h$  is the smoothing distance. The approach is in keeping with the meshless techniques in general and its implementation in 3D is not complex. The contact algorithm allows surfaces to come together and separate in a physically correct manner.

## 4.5 Validation Tests

### 4.5.1 Wave propagation test

The first test with which to validate the implementation of the coupling algorithm between DYNA3D and the SPH solver is the same wave propagation test from [ 80 ] that has been used in the previous chapter to test the SPH contact algorithm. Only the second variation on this test will be presented in this paragraph, namely the test with a block wave travelling through the bars. To recapitulate, the test setup is as follows: two steel bars of 50cm length are initially in contact and one free end of the bars is subjected to a prescribed velocity of 0.001cm/ $\mu$ s. The material properties are given in Table 3.1.

Four simulations were performed, an FE only (with a DYNA3D penalty contact algorithm), an SPH only, and two coupled FE-SPH simulations. In one of the coupled simulations the prescribed velocity is applied to the finite elements. This means that the wave first travels through the bar discretised with finite elements, and the wave is then transferred to the bar discretised with SPH particles via the coupling algorithm. In the second coupled simulation the prescribed velocity is applied to the bar discretised with SPH particles.

In order to simulate this 1D problem in DYNA3D a mesh with symmetry planes is required. A volume of 5x5x50 cm was meshed for each bar, resulting in a mesh of 4x4x50 elements. The symmetry planes were applied along the length of the bars, thus ensuring a 1D wave propagation.

The SPH only simulation in 3D obviously also requires the use of symmetry planes. The symmetry planes are implemented using a ghost particle approach [ 77 ]. A finer discretisation was used here to avoid a direct influence of the symmetry planes on the central particles. Again a volume of 5x5x50 is discretised. The total number of particles in each bar is 10000 (10x10x100). The same linear elastic material model was used as in the FE simulation. In this case two simulations were run, one without contact (Figure 4.9) and one with contact (Figure 4.10).

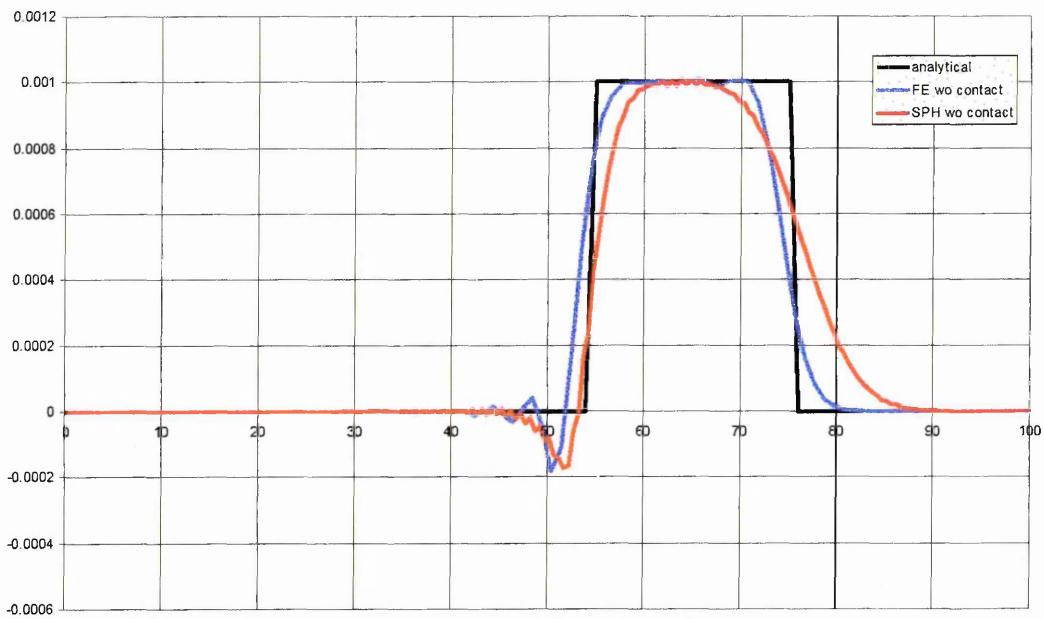


Figure 4.9: FE and SPH (without contact) Velocity Profiles of 3D Wave Propagation Simulation, at 128.2 $\mu$ s.

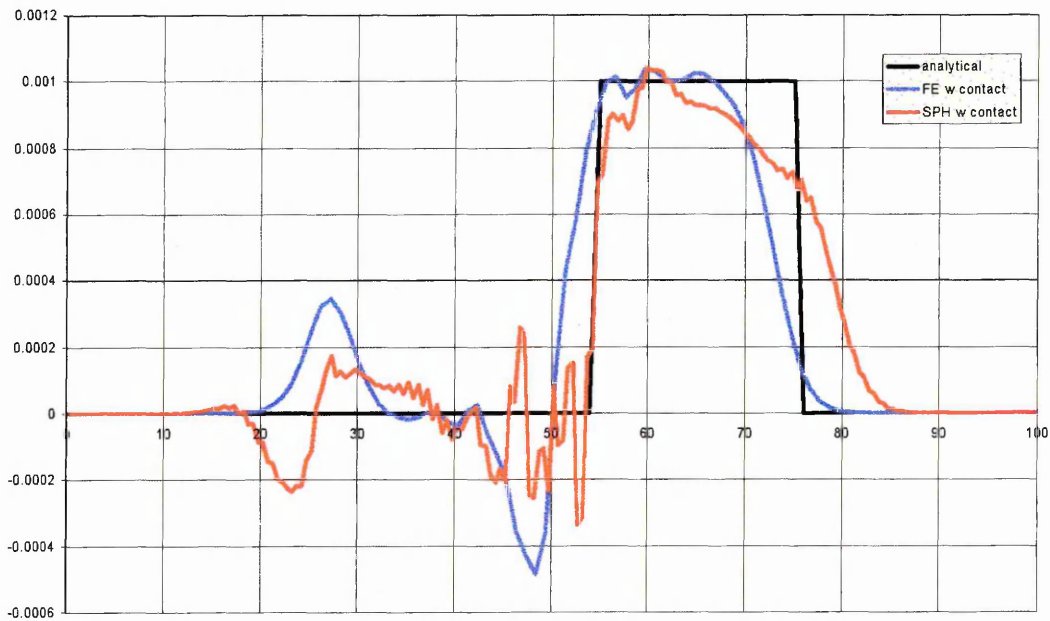


Figure 4.10: FE and SPH (with contact) Velocity Profiles of 3D Wave Propagation Simulation, at 128.2 $\mu$ s.

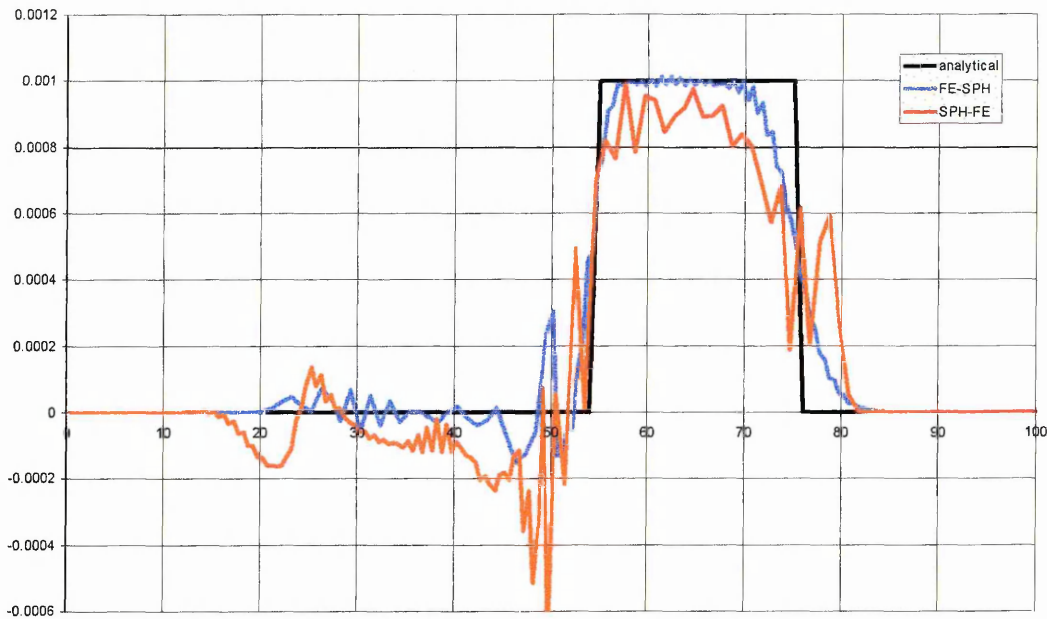


Figure 4.11: Coupled FE-SPH and SPH-FE Velocity Profiles of 3D Wave Propagation Simulation, at  $128.2\mu\text{s}$ .

By comparing the results of the FE only and SPH only, with and without contact, one can immediately see that the introduction of a contact interface causes a big change in the velocity profile. The introduction of a contact algorithm causes a change in the distance the wave propagates into the second bar, a reflected wave into the first bar and oscillations around the contact surface. These features can be observed in the SPH only with potential force contact algorithm as well as FE only with master-slave contact algorithm. The magnitude of these spurious effects is similar for both sets of results. This demonstrates that the potential force contact algorithm performs similarly to a penalty stiffness contact in an FE code.

Figure 4.11 shows the results obtained with the coupled FE-SPH code. The velocity profile labelled FE-SPH is the result obtained when the block wave first travels through the FE mesh. The second curve, labelled SPH-FE, shows the results when the wave travels in the SPH part first. The simulations show a spurious pressure wave travelling from the contact surface.

#### 4.5.2 Plate Impact

A second test that was used to validate the coupling between FE and SPH using the potential force contact algorithm is a symmetric bar impact. The problem setup is as follows: two steel plates of 50cm length are initially in contact and are travelling with an equal and opposite velocity of  $0.02\text{cm}/\mu\text{s}$  (ie.  $200\text{m/s}$ ) towards each other. Symmetry planes are applied along the sides of the bars, this results in a 1D wave propagation in the plates. A volume of  $5\times 5\times 50\text{ cm}$  was discretised for each plate. A linear elastic material model was used as in the FE simulation. Three simulations were performed, an FE only (with a DYNA3D penalty contact algorithm), an SPH only to provide a point of comparison, and a coupled FE-SPH simulation.

The resulting velocity profiles in the plates are shown in Figure 4.12 and Figure 4.13 at times  $48\mu\text{s}$  and  $300\mu\text{s}$  respectively. At  $48\mu\text{s}$  (Figure 4.12) the plates are in contact and a stress wave is travelling away from the contact point. It can be seen that the contact algorithm performs very well. The velocity behind the stress wave front is zero. The level of oscillation of the SPH and coupled FE-SPH results are lower than the oscillations of the DYNA3D FE simulation. This is probably due to the non-linear nature of the contact potential. This ensures a gradual increase of the contact forces and avoids the ‘shatter’ of the contact surface which is sometimes observed in penalty contact algorithms used in FE codes. There are small differences in the distance the waves have travelled. This is due to the fact that the potential contact algorithm detects contact when two nodes are within each other kernel support. Because FE nodes are on the boundary of the discretised domain contact will be detected slightly early when using the potential contact algorithm, hence the wave has travelled slightly further into the bars in this case. The stress wave of the SPH simulation has advanced a slightly smaller distance in the plates. But it can be seen that the point at which the velocity starts to decrease starts at the same point as the FE simulation. However the slope of the velocity profile is shallower than in the FE results. This is due to the non-linear nature of the potential contact algorithm which means that the contact force increases gradually as the plates approach each other. At  $300\mu\text{s}$  (Figure 4.13) the release waves have reached the contact point and the plates are travelling away from each other. A small amount of oscillation can be observed in the SPH and SPH-FE results. This oscillations are a results of the instabilities in the SPH method and not related to the contact algorithm or coupling algorithm.

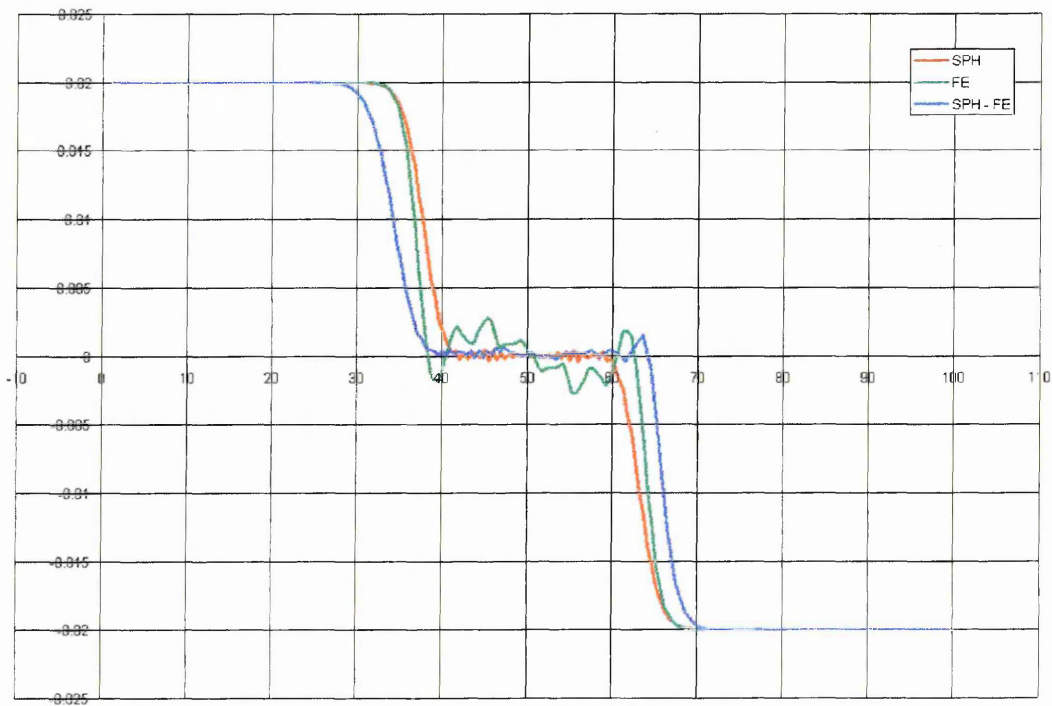


Figure 4.12: Plate Impact Velocity Profiles at  $48\mu\text{s}$ .

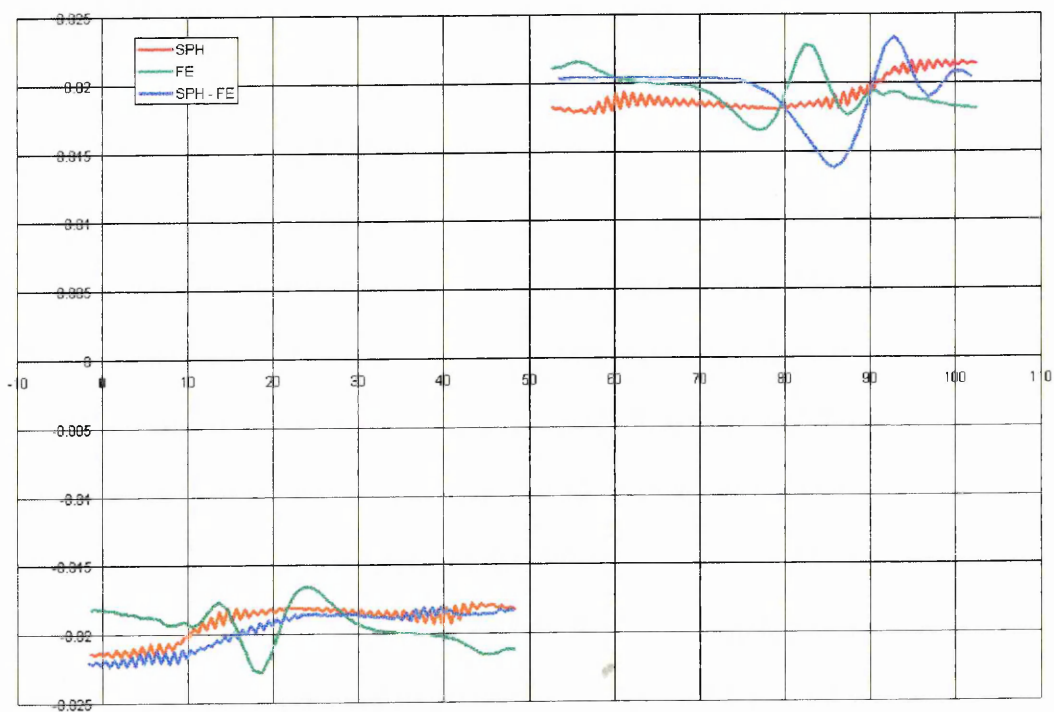


Figure 4.13: Plate Impact Velocity Profiles at 48μs.

## Chapter 5 Anisotropic Plasticity

### 5.1 Introduction

In order to accurately predict the behaviour of structural components under impact loading it is necessary to use a material model that can take account of the material characteristics that influence the response of the structure. One has to be able to model elastic behaviour, hardening, rate dependency, and thermal softening. Materials can be anisotropic with respect to all these effects. The purpose of this chapter is to present the underlying theory and implementation of a material model that is capable of dealing with:

- ▶ material anisotropy, both in elastic and plastic regimes
- ▶ isotropic - and kinematic hardening
- ▶ strain rate effects
- ▶ non-linear pressure dependency (EOS)
- ▶ tensile failure

At this stage thermal softening will not be taken into account. The materials of primary interest are materials used in aircraft crashworthiness, and frequently exhibit one or more of the above characteristics. For example, an aluminium alloy will in general be cold worked so it can exhibit a certain degree of anisotropy. Because of the nature of the cold working operations, rolling or extrusion, the material can be assumed to behave in an orthotropic fashion. As a consequence orthotropic elastic and plastic constitutive relationships should be used if the material is to be represented accurately. Because of the orthotropic plastic behaviour the formulation of the yield condition will be affected. In general there can be several material characteristics that lead to anisotropic behaviour:

- ▶ The orientation of single-crystal materials.
- ▶ Microstructural directionality, for example grain shape.
- ▶ Long-range microstructural features such as in reinforced composites.

Cyclic plasticity effects will generally not be pronounced during an impact. So it is not necessary to consider material models which are capable of modelling complex cyclic loading. However when dealing with the collapse of structures or components it is possible for the load path to significantly change during the impact. Therefore it would be useful that the Bauschinger effect could be taken into account, as the difference in the yield stress on load reversal can be quite significant.

Another requirement for the material model is that it must be capable of representing the material behaviour over a range of strains and strain rates. In crashworthiness applications typically from  $10^{-3} \text{ s}^{-1}$  to  $300 \text{ s}^{-1}$ .

In the following sections tensor notation has been used for most of the equations, however where it was required for clarity, Voigt notation was used.

## 5.2 Theory

### 5.2.1 Homogeneous Functions

The plasticity model derived in this chapter is based on the assumption that certain functions are homogenous. To this end, it is worth noting that a homogeneous function of degree  $n$  is defined as a function that satisfies the following condition (Euler's theorem):

$$f(\lambda x_1, \dots, \lambda x_i, \dots) = \lambda^n f(x_1, \dots, x_i, \dots), \quad (\text{Eq. 5.1})$$

and a direct consequence of this is that:

$$\frac{\partial f}{\partial x} : x = n f(x) \quad (\text{Eq. 5.2})$$

### 5.2.2 Yield Criterion

The first problem that has to be addressed when modelling plastic behaviour of materials is the identification of the point at which the material starts to yield ie. accumulate permanent deformation. This point is defined by a yield criterion. The most commonly used yield criteria are Von Mises' yield criterion and Tresca's yield criterion. Von Mises' yield criterion is:

$$f(s) = \sqrt{s:s} - R = 0 \quad (\text{Eq. 5.3})$$

where

$$R = \sqrt{\frac{3}{2}} \sigma_{\text{Yield}}$$

and  $s$  is the deviatoric stress tensor. It expresses the assumption that yield will occur once the distortional energy has reached the same level as when the yield stress is reached in simple tension. This criterion is isotropic, this means that yielding will start at the same point irrespective of the orientation of the material.

Tresca's yield criterion can be expressed as:

$$f = \max_{i \neq j} (|\sigma_i - \sigma_j|) - \sigma_Y = 0, \quad (\text{Eq. 5.4})$$

and states that yielding will occur once the maximum shear stress reaches the same value as the yield stress in uniaxial tension. Again, this criterion is isotropic.

When dealing with materials that exhibit different yield stresses in different directions one needs to use more complex yield function that allows for this directional

dependency of the yield stress. Several functions have been proposed that are basically extensions of the Von Mises' yield criterion.

In general one could introduce a yield function following Lemaitre [ 57 ]:

$$(A_{ijkl} : s_{kl}) : s_{ij} = 1 \quad (\text{Eq. 5.5})$$

where A is a fourth order tensor with the following symmetries:

$$A_{ijkl} = A_{klij} = A_{jikl} = A_{ijkl} \quad (\text{Eq. 5.6})$$

Using this notation the yield surface can be expressed as:

$$f(\sigma_{ij}) = \sqrt{(A : s) : s} - R = 0 \quad (\text{Eq. 5.7})$$

And the equivalent stress is:

$$\bar{\sigma} = \sqrt{\frac{3}{2} (A : s) : s}$$

The indices have been omitted for brevity.

The anisotropic yield criterion proposed by Hill [ 44 ] fits into this category, and will be the starting point of the material model in this chapter. It can be expressed as follows:

$$f(\sigma_{ij}) = \frac{1}{\sqrt{F+G+H}} \left[ F(\sigma_y - \sigma_z)^2 + G(\sigma_z - \sigma_x)^2 + H(\sigma_x - \sigma_y)^2 + 2L\sigma_{yz}^2 + 2M\sigma_{zx}^2 + 2N\sigma_{xy}^2 \right]^{\frac{1}{2}} - R = 0 \quad (\text{Eq. 5.8})$$

In the plane stress case this reduces to:

$$\left[ \frac{F\sigma_y^2 + G\sigma_x^2 + H(\sigma_x - \sigma_y)^2 + 2N\sigma_{xy}^2}{F+G+H} \right]^{\frac{1}{2}} - R = 0 \quad (\text{Eq. 5.9})$$

In this formulation it is assumed that the anisotropy has three mutually orthogonal planes of symmetry at every point in the material. The intersections of these planes are the principal axes of anisotropy. The same yield criterion also be expressed by using Lankford's coefficients ie.

$$F = \frac{1}{R+1} \frac{R}{P} \quad (\text{Eq. 5.10})$$



$$G = \frac{1}{R+1} \quad (\text{Eq. 5.11})$$

$$H = \frac{1}{R+1} R \quad (\text{Eq. 5.12})$$

$$2L = (2Q_{yz} + 1) \quad (\text{Eq. 5.13})$$

$$2M = (2Q_{zx} + 1) \frac{PR + R}{P(R+1)} \quad (\text{Eq. 5.14})$$

$$2N = (2Q_{xy} + 1) \frac{P + R}{P(R+1)} \quad (\text{Eq. 5.15})$$

the yield function can be rewritten as:

$$\begin{aligned} f(\sigma_{ij}) = & \sqrt{R+P+RP} \left[ R(\sigma_y - \sigma_z)^2 + P(\sigma_z - \sigma_x)^2 + PR(\sigma_x - \sigma_y)^2 \right. \\ & + (2Q_{yz} + 1)(PR + P)\sigma_{yz}^2 + (2Q_{zx} + 1)(PR + R)\sigma_{zx}^2 \\ & \left. + (2Q_{xy} + 1)(P + R)\sigma_{xy}^2 \right]^{\frac{1}{2}} - R = 0 \end{aligned} \quad (\text{Eq. 5.16})$$

Alternatively Eq. 5.7 can also be expressed in terms of the full stress tensor:

$$f(\sigma) = \sqrt{(A' : \sigma) : \sigma} - R = 0 \quad (\text{Eq. 5.17})$$

The parameter  $R$  in this case is the yield stress along one of the principal axes of anisotropy, and is used in order not to cause any confusion with the anisotropy parameter  $R$ . The same equation can be written in Voigt notation as:

$$f(s) = \sqrt{(A s) : s} - R = 0 \quad (\text{Eq. 5.18})$$

where  $A$  is a 6x6 matrix and  $s$  is a 6x1 column vector.  $A$  takes the same shape of an orthotropic compliance matrix ie.

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & 0 & 0 & 0 \\ A_{21} & A_{22} & A_{23} & 0 & 0 & 0 \\ A_{31} & A_{32} & A_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & A_{66} \end{bmatrix} \quad (\text{Eq. 5.19})$$

If the terms in the stress vector are written in the following order:

$$s^t = \begin{bmatrix} s_{xx} & s_{yy} & s_{zz} & s_{yz} & s_{xz} & s_{xy} \end{bmatrix} \quad (\text{Eq. 5.20})$$

using these two expressions, substituting in Eq. 5.18, rearranging terms and comparing with Eq. 5.8 results in the following expressions for the coefficients of A:

$$A = \frac{1}{F+G+H} \begin{bmatrix} G+H & -H & -G & 0 & 0 & 0 \\ -H & F+H & -F & 0 & 0 & 0 \\ -G & -F & F+G & 0 & 0 & 0 \\ 0 & 0 & 0 & 2N & 0 & 0 \\ 0 & 0 & 0 & 0 & 2L & 0 \\ 0 & 0 & 0 & 0 & 0 & 2M \end{bmatrix} \quad (\text{Eq. 5.21})$$

For an isotropic material, the material constants are:

$$F=G=H=\frac{1}{2} \quad (\text{Eq. 5.22})$$

$$L=M=N=\frac{3}{2} \quad (\text{Eq. 5.23})$$

Using these values Hill's yield criterion reduces to Von Mises'.

In Eq. 5.18 and Eq. 5.7 the A matrices are denoted as A and A', but in the case of Hill's yield function these two matrices are the same. This is obvious as all terms containing diagonal terms of the stress tensor are differences of two components.

### 5.2.3 Flow Rule

The next issue to be addressed is the evolution of the plastic strain. In the current context the assumption will be made that the total strain can be decomposed (additive decomposition) into elastic and plastic parts:

$$\varepsilon = \varepsilon^{el} + \varepsilon^{pl} \quad (\text{Eq. 5.24})$$

If a material is stressed such that the material is loaded beyond its plastic limit, then the plastic strain will increase. The flow rule relates the change of the plastic strain tensor to the change of stress state. Generally it is assumed that there is a plastic potential Q which is a function of the stress tensor and hardening variables [ 57 ]. The evolution of the plastic strain is assumed to be proportional to the gradient of this plastic potential Q:

$$\dot{\varepsilon}^{pl} = \dot{\lambda} \frac{\partial Q}{\partial \sigma} \quad (\text{Eq. 5.25})$$

This is the Prandtl-Reuss flow rule [ 57 ]. This equation assumes that the increment in the plastic strain tensor is in the direction of the normal to the plastic potential  $Q$ . A variation on this is the Mises-Levy flow rule [ 57 ] which relates the total strain, rather than the plastic strain, to the gradient of the plastic potential:

$$\dot{\epsilon} = \dot{\lambda} \frac{\partial Q}{\partial \sigma}. \quad (\text{Eq. 5.26})$$

A special case of this theory is the associative plasticity where the yield function is equal to the plastic potential ie.  $f = Q$ . The general case where  $f \neq Q$  is known as non-associative plasticity.

#### 5.2.4 Hardening

The hardening rule is the equation that describes how the yield surface changes during plastic deformation. The two most commonly used hardening rules are isotropic hardening and kinematic hardening. Isotropic hardening is a uniform expansion of the yield surface due to plastic deformation. The variable used to measure the plastic deformation is either the effective plastic strain  $\bar{\epsilon}^{pl}$  or the plastic work  $W_p$ . The case where the effective plastic strain is used to calculate the increase in size of the yield surface  $R$  is called Strain Hardening. If the plastic work is used the term Work Hardening is used.

The evolution of the radius of the yield surface is assumed to be proportional to the measure for the plastic deformation:

$$dR = C_i d\bar{\epsilon}^{pl}, \quad (\text{Eq. 5.27})$$

or in rate form:

$$\dot{R} = C_i \dot{\bar{\epsilon}}^{pl}. \quad (\text{Eq. 5.28})$$

The value of the constant  $C_i$  describing the isotropic hardening behaviour will be determined next.

In the case of pure isotropic hardening the effective stress is

$$\bar{\sigma} = \sqrt{\frac{3}{2} (A : \sigma) : \sigma} \quad (\text{Eq. 5.29})$$

According to the yield criterion Eq. 5.7, we get

$$R = \sqrt{\frac{2}{3} \bar{\sigma}}. \quad (\text{Eq. 5.30})$$

Taking time derivatives results in:

$$\dot{R} = \sqrt{\frac{2}{3}} \dot{\bar{\sigma}} \quad (\text{Eq. 5.31})$$

The stress is related to plastic strain by the hardening modulus  $H'$ . Expressed in rate form this is:

$$\dot{\sigma}_{ij} = H' \dot{\epsilon}_{ij}^{pl} \quad (\text{Eq. 5.32})$$

The same expression holds for the relation between the equivalent stress and the equivalent plastic strain (if  $f$  is homogeneous):

$$\dot{\bar{\sigma}} = H' \dot{\bar{\epsilon}}^{pl}$$

Combining this with Eq. 5.31 the rate of expansion of the yield surface radius becomes

$$\dot{R} = \sqrt{\frac{2}{3}} H' \dot{\bar{\epsilon}}^{pl} \quad (\text{Eq. 5.33})$$

and we obtain the constant  $C_i$ :

$$C_i = \sqrt{\frac{2}{3}} H'. \quad (\text{Eq. 5.34})$$

Kinematic Hardening consists of shifting the initial yield surface in the stress space. The shape and size of the initial yield surface is maintained. This is achieved by introducing backstress  $\alpha$ , which is the shift of the yield surface in the deviatoric stress space.

In the case of linear kinematic hardening, the assumption made by Prager is that the translation of the yield surface is in the direction of the plastic strain-rate tensor ie.

$$\dot{\alpha} = C_k \dot{\epsilon}^{pl} \quad (\text{Eq. 5.35})$$

In 1D the movement of the yield surface is equal to the difference between the initial yield stress  $\sigma_{Y_0}$  and the current effective stress  $\bar{\sigma}$ .

The plastic strain-rate tensor is:

$$\dot{\epsilon}^{pl} = \dot{\lambda} \frac{\partial f}{\partial \sigma} \quad (\text{Eq. 5.36})$$

Hence,

$$\dot{\alpha} = C_k \dot{\lambda} \frac{\partial f}{\partial \sigma}$$

In APPENDIX C it is shown that the following relationship holds:

$$\dot{\bar{\epsilon}}^{pl} = \sqrt{\frac{2}{3}} \dot{\lambda}$$

or

$$\dot{\lambda} = \sqrt{\frac{3}{2}} \dot{\bar{\epsilon}}^{pl}. \quad (\text{Eq. 5.37})$$

Hence the expression for the rate of change of  $\alpha$  can be written as:

$$\dot{\alpha} = C_k \sqrt{\frac{3}{2}} \dot{\bar{\epsilon}}^{pl} \frac{\partial f}{\partial \sigma} \quad (\text{Eq. 5.38})$$

Where

$$\frac{\partial f}{\partial \sigma} = \frac{A : \xi}{\sqrt{(A : \xi) : \xi}} = \sqrt{\frac{3}{2}} \frac{A : \xi}{\bar{\sigma}} \quad (\text{Eq. 5.39})$$

The unit normal  $N$  to  $f$  is:

$$N = \frac{A : \xi}{\sqrt{(A : \xi) : (A : \xi)}} = \frac{\partial f}{\partial \sigma} \frac{\sqrt{(A : \xi) : (A : \xi)}}{\sqrt{(A : \xi) : \xi}}$$

which results in:

$$\dot{\bar{\alpha}} = \sqrt{(A : \dot{\alpha}) : \dot{\alpha}} = C_k \sqrt{\frac{3}{2}} \dot{\bar{\epsilon}}^{pl} \sqrt{\left(A : \frac{\partial f}{\partial \sigma}\right) : \frac{\partial f}{\partial \sigma}}$$

$$\dot{\bar{\alpha}} = C_k \sqrt{\frac{3}{2}} \frac{\dot{\bar{\sigma}}}{H'} \sqrt{\left(A : \frac{\partial f}{\partial \sigma}\right) : \frac{\partial f}{\partial \sigma}}$$

In the 1D case where  $\dot{\bar{\alpha}} = \dot{\bar{\sigma}}$  the constant  $C_k$  becomes

$$C_k = \sqrt{\frac{2}{3}} \frac{1}{\sqrt{\left(A : \frac{\partial f}{\partial \sigma}\right) : \frac{\partial f}{\partial \sigma}}} H'$$

One can see that in the case of Von Mises plasticity this equation reduces to:

$$C_k = \frac{2}{3} H'. \quad (\text{Eq. 5.40})$$

If a combination of isotropic and kinematic hardening is used a factor  $\beta$  is introduced to define the relative share of the isotropic and kinematic part of hardening (see Eq. 5.41 and Eq. 5.42). The ratio can be determined by performing a load reversal test. The point at which yielding starts under reversed load in this test can be used to calculate this factor  $\beta$ .

$$\dot{R} = \beta C_i \dot{\epsilon}^{pl} \quad (\text{Eq. 5.41})$$

$$\dot{\alpha} = (1 - \beta) C_k \dot{\epsilon}^{pl} \quad (\text{Eq. 5.42})$$

### 5.2.5 Determination of the parameter $\lambda$

To complete the description of the plasticity model the parameter  $\lambda$  will be calculated. Starting from the consistency condition:

$$\dot{f} = \frac{\partial f}{\partial \sigma} : \dot{\sigma} + \frac{\partial f}{\partial \alpha} : \dot{\alpha} + \frac{\partial f}{\partial R} \dot{R} = 0 \quad (\text{Eq. 5.43})$$

and the yield surface:

$$f(\sigma, \alpha, R) = \sqrt{(A : \xi) : \xi} - R = 0$$

the anti-symmetry in  $\sigma_{ij}$  and  $\alpha_{ij}$  and the fact that:

$$\frac{\partial f}{\partial R} = -1$$

the consistency condition can be rewritten as:

$$\dot{f} = \frac{\partial f}{\partial \sigma} : \dot{\sigma} - \frac{\partial f}{\partial \alpha} : \dot{\alpha} - \dot{R} = 0 \quad (\text{Eq. 5.44})$$

Using

$$\dot{\sigma} = C(\dot{\epsilon} - \dot{\epsilon}^{pl})$$

and Eq. 5.41 and Eq. 5.42 the expression for  $\dot{f}$  can be rewritten as:

$$\dot{f} = \frac{\partial f}{\partial \sigma} : [C : (\dot{\epsilon} - \dot{\epsilon}^{pl})] - (1 - \beta) C_k \frac{\partial f}{\partial \sigma} : \dot{\epsilon}^{pl} - \beta C_i \dot{\epsilon}^{pl} = 0 \quad (\text{Eq. 5.45})$$

In order to eliminate the effective plastic strain in the above equation we can use

$$\dot{\epsilon}^{pl} = \sqrt{\frac{2}{3}} \dot{\lambda}$$

Using this result and the fact that

$$\dot{\epsilon}^{pl} = \dot{\lambda} \frac{\partial f}{\partial \sigma} \quad (\text{Eq. 5.46})$$

the consistency equation can be rewritten as:

$$\dot{f} = \frac{\partial f}{\partial \sigma} : (C : \dot{\epsilon}) - \frac{\partial f}{\partial \sigma} : (C : \frac{\partial f}{\partial \sigma}) \dot{\lambda} - (1 - \beta) C_k \frac{\partial f}{\partial \sigma} : \frac{\partial f}{\partial \sigma} \dot{\lambda} - \sqrt{\frac{2}{3}} \beta C_i \dot{\lambda} = 0 \quad (\text{Eq. 5.47})$$

This expression can now be solved for  $\dot{\lambda}$  to obtain the following expression:

$$\dot{\lambda} = \frac{\frac{\partial f}{\partial \sigma} : (C : \dot{\epsilon})}{\frac{\partial f}{\partial \sigma} : (C : \frac{\partial f}{\partial \sigma}) + (1 - \beta) C_k \frac{\partial f}{\partial \sigma} : \frac{\partial f}{\partial \sigma} + \sqrt{\frac{2}{3}} \beta C_i} \quad (\text{Eq. 5.48})$$

### 5.2.6 Strain-Rate dependency

Many research workers have reached independently the same conclusion that some metals such as mild steel are very sensitive to strain-rate. A dependency on strain rate of the yield stress and the hardening modulus can be observed.

One of the first models that reasonable successfully took into account strain rate effects is the Cowper-Symonds model [ 50 ]:

$$\sigma_{\text{eff}} = \sigma_{\text{eff},0} \left( 1 + \left( \frac{\dot{\epsilon}}{B + C\epsilon} \right)^{1/q} \right) \quad (\text{Eq. 5.49})$$

Further developments was made in the Johnson Cook model and the Zerilli-Armstrong model.

Johnson-Cook [ 48 ], [ 49 ]:

$$\sigma_{\text{eff}} = (A + B\epsilon^n) (1 + C \ln \dot{\epsilon}^*) (1 - T^{*m}) \quad (\text{Eq. 5.50})$$

Zerilli-Armstrong [ 89 ]:

$$\sigma_{\text{eff}} = C_0 + C_2 \epsilon^{1/2} \exp(-C_3 T + C_4 T \ln \dot{\epsilon}) \quad (\text{fcc metals}) \quad (\text{Eq. 5.51})$$

$$\sigma_{\text{eff}} = C_0 + C_1 \exp(-C_3 T + C_4 T \ln \dot{\epsilon}) + C_5 \epsilon^n \quad (\text{bcc metals}) \quad (\text{Eq. 5.52})$$

The Cowper-Symonds and Johnson-Cook models are merely curve fitted models. There is no real physical background to the function proposed to describe the strain rate effect. This is where the Zerilli-Armstrong model differs. It is based on dislocation mechanics, and a thermal activation analysis. This results in different expression for the dynamic yield stress depending on crystal structure. The Zerilli-Armstrong model is one of the first material models that takes into account the crystal structure of the material. Different expressions are used for face centred cubic and body centred cubic crystal structures. All of these models are based on a power strain hardening law, i.e. the stress increases according to a power function with respect to the strain.

### 5.2.7 Equation of State

The behaviour of materials under general three-dimensional stress states is usually modelled by decomposing the stress into hydrostatic components and deviatoric parts. An equation of state for a material is a mathematical formulation which describes the relationship between the hydrostatic components of stress and strain. Since high rate deformation generates high temperatures, the effect of temperature or internal energy on pressure must be taken into account. An equation of state is a three dimensional constitutive relation which defines the states that a material can achieve, and can generally be expressed as:

$$p = f(E, V), \quad (\text{Eq. 5.53})$$

where  $p$  is the pressure,  $E$  is the internal energy and  $V$  is the relative volume.

One of the most commonly used equations to model material behaviour under impact conditions is the Gruneisen equation of state. It defines the pressure for compressed materials ( $\mu > 0$ ) as:

$$p = \frac{\rho_0 C_0^2 \mu \left[ 1 + \left( 1 - \frac{\Gamma}{2} \right) \mu - \frac{a}{2} \mu^2 \right]}{\left[ 1 - (S_1 - 1) \mu - S_2 \frac{\mu^2}{\mu + 1} - S_3 \frac{\mu^3}{(\mu + 1)^2} \right]} + (\Gamma + a \mu) E \quad (\text{Eq. 5.54})$$

and for expanded materials ( $\mu < 0$ ) as:

$$p = \rho_0 C_0^2 \mu + (\Gamma + a \mu) E, \quad (\text{Eq. 5.55})$$

where  $C_0$  is the intercept of the shock velocity vs. particle velocity ( $v_s - v_p$ ) curve,  $S_1$ ,  $S_2$ ,  $S_3$  are coefficients of the slope of the ( $v_s - v_p$ ) curve,  $\Gamma$  is Grüneisen gamma.



This equation is valid for single shocked materials without a change of phase [ 94 ].

### 5.2.8 Spall

Spall is a mode of failure of materials under hydrostatic tension. One definition of this phenomenon is [ 94 ] ‘Spall is material failure due to the interaction between two or more rarefaction waves’. In hydrocodes, the main ways to model spall failure are:

- ▶ Pressure cut-off
- ▶ Maximum principal stress
- ▶ Grady spall criterion

These models will briefly be discussed here because they will be implemented in the material model. The first two models are very simple. The pressure cut-off model simply compares the pressure with a user defined pressure cut-off value. Once the pressure is less than this pressure cut-off value the material is assumed to have spalled. The deviatoric stress tensor and the pressure are set to zero, and no hydrostatic tension is subsequently allowed. The maximum principal stress criterion is check whether the maximum principal stress has exceeded a user defined cut-off value. Once spall is detected the deviatoric stress tensor and the pressure are again set to zero, and no hydrostatic tension is subsequently allowed.

The Grady spall model is energy based and assumes the material spalls when the strain energy reaches a certain level [ 33 ]. Grady checks how much energy is required to be able to create spalled material. Two mechanisms are considered, brittle fracture and ductile fracture. The energy that is required in the case of brittle fracture is based on the critical fracture toughness. In the case of ductile fracture it is based on work required to reach the failure strain. This method then results in the following spall strengths:

$$p_s = \sqrt{2\rho c_0^2 \sigma_Y \epsilon_{fail}} \quad \text{Ductile Failure} \quad (\text{Eq. 5.56})$$

$$p_s = \sqrt[3]{3\rho c_0 K_c^2 \dot{\epsilon}} \quad \text{Brittle Failure} \quad (\text{Eq. 5.57})$$

There is a transition point between ductile and brittle spall for a certain strain rate. This critical strain rate can be calculated from:

$$\dot{\epsilon}_{crit} = \sqrt{\frac{8 B_0^2 (\sigma_Y \epsilon_{fail})^3}{9 \rho K_c^4}} \quad (\text{Eq. 5.58})$$

## 5.3 Material Testing

The determination of the material constants for an anisotropic material is more complex than for an isotropic material. Several types of tests can be conducted. Hill proposes a series of simple tensile tests in [ 44 ]. If one determines the yield stress of the material in the three principal axes of anisotropy then it is clear that:

$$\begin{aligned}
\frac{1}{\sigma_{Y,x}} &= G + H, \\
\frac{1}{\sigma_{Y,y}} &= H + F, \\
\frac{1}{\sigma_{Y,z}} &= F + G,
\end{aligned}
\tag{Eq. 5.59}$$

from which one could determine F, G and H as follows:

$$\begin{aligned}
2F &= \frac{1}{\sigma_{Y,y}} + \frac{1}{\sigma_{Y,z}} - \frac{1}{\sigma_{Y,x}} \\
2G &= \frac{1}{\sigma_{Y,z}} + \frac{1}{\sigma_{Y,x}} - \frac{1}{\sigma_{Y,y}} \\
2H &= \frac{1}{\sigma_{Y,x}} + \frac{1}{\sigma_{Y,y}} - \frac{1}{\sigma_{Y,z}}
\end{aligned}
\tag{Eq. 5.60}$$

The other three material constants for Hill's yield criterion can be determined from the yield stresses in shear with respect to the principal axes of anisotropy.

$$\begin{aligned}
2L &= \frac{1}{\tau_{Y,y}} \\
2M &= \frac{1}{\tau_{Y,z}} \\
2N &= \frac{1}{\tau_{Y,x}}
\end{aligned}
\tag{Eq. 5.61}$$

However instead of this direct method of determining the anisotropy parameters, it is better to measure the incremental strains in all three directions during a tensile test. For example for a tensile test in the X-direction these strains would be in the ratios:

$$d\epsilon_x : d\epsilon_y : d\epsilon_z = G+H : -H : -G \tag{Eq. 5.62}$$

Similarly tensile tests in the y and z directions will provide the ratios between F and H, and G and F. This then allows a test of the theory as the following identity should be satisfied:

$$\left(\frac{H}{G}\right)\left(\frac{G}{F}\right)\left(\frac{F}{H}\right) = 1 \tag{Eq. 5.63}$$

This method is preferable to the direct method if the yield stress is not very well defined. It can also be used as a way to determine the through thickness yield stress when the material is in the form of a thin sheet.

Another test that can be performed to characterise material anisotropy is a punch-stretching test in a hydraulic press [ 75 ]. This test is particularly useful when data is required about thin sheets of metal. The test consists of applying a stretching force to a sheet of metal by means of a rigid semi-spherical punch (see Figure 5.1). The test specimen geometry can be circular or a thin strip of material that is clamped at the edges. On the sheet metal surface a grid of circles is etched or photo-engraved (see Figure 5.2 and Figure 5.3). As the material deforms, so will the circles. From the deformed shape of the circles one can then determine in-plane strains. The main difficulty lies in ensuring a 'frictionless contact between the punch and the test specimen.

From this test one can determine the flow curves of the material. It also enables the determination of the strain-rate dependency of these flow curves. When the tests are conducted on thin strips of metal the influence of material anisotropy can be determined. In order to achieve this the tests are performed on strips cut at various angles with the rolling direction of the sheet. For example, a series of tests with strips cut at angle of 0, 45 and 90 degrees with the rolling direction will allow the in-plane anisotropy to be characterised.

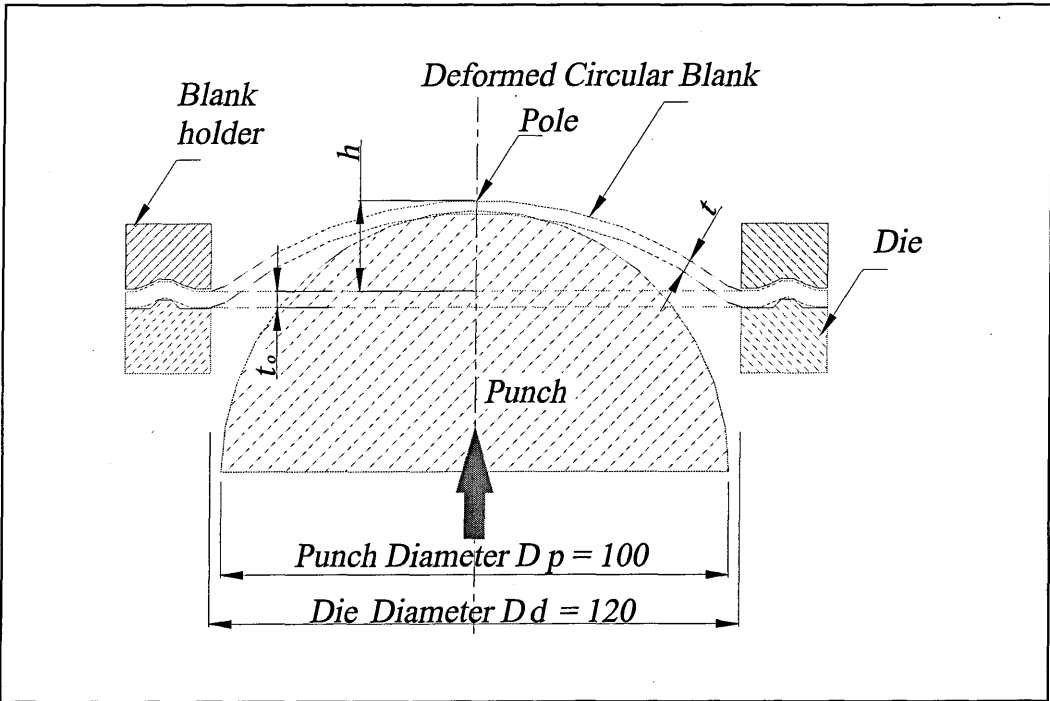


Figure 5.1: Punch Stretching Arrangement [ 75 ].

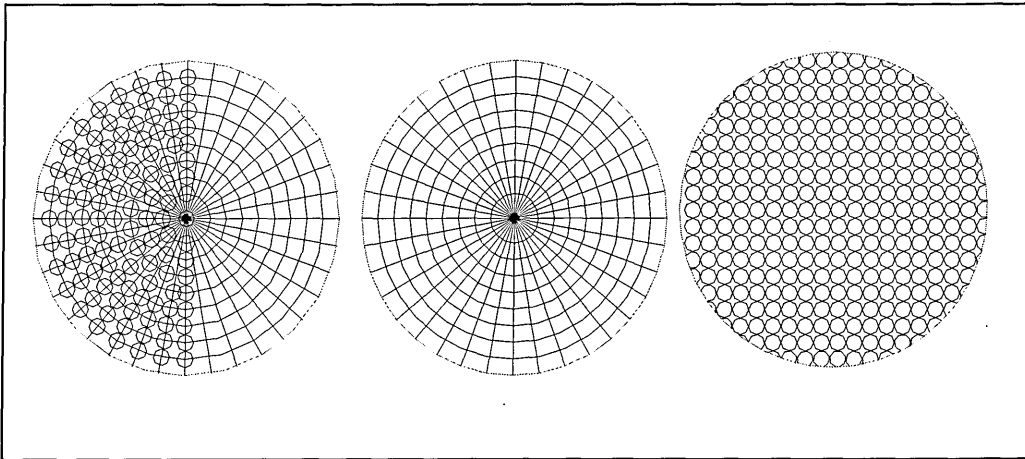


Figure 5.2: Different Types of Marking of the Bulge Test Specimens [ 75 ].

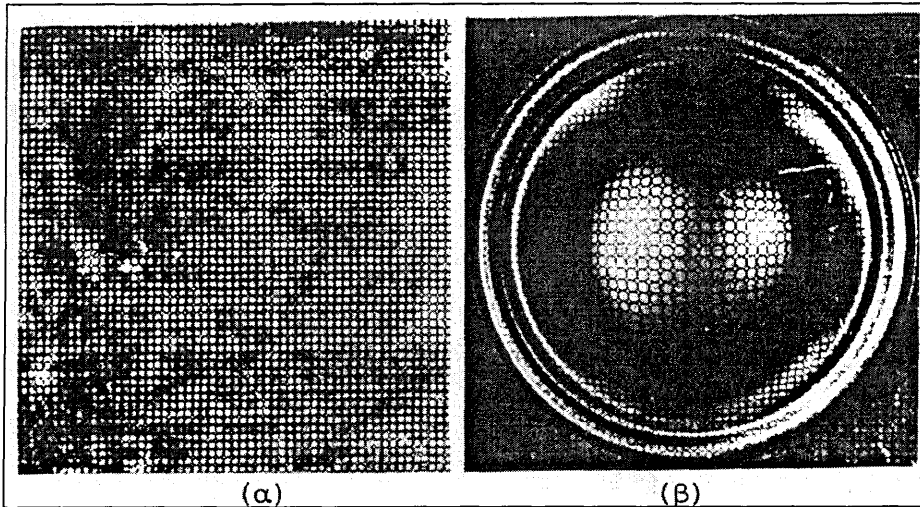


Figure 5.3: Photos of the Test Specimens Marking before and after the test [ 75 ].

#### 5.4 Implementation in LLNL-DYNA3D

This paragraph will explain how a model that takes account of the above introduced theoretical concepts was implemented into an explicit finite element code, Cranfield University's version of LLNL-DYNA3D.

##### 5.4.1 Overview of LLNL-DYNA3D F3DM33 routine

The model used as a basis for implementing the theory discussed in section 5.2 is the basic anisotropy model that is available in Cranfield University's version of public domain DYNA3D, which was originally developed at Lawrence Livermore National Laboratories [ 61 ]. [This model is a general anisotropic elastic-plastic material model.] It [combines the orthotropic elastic model of material type 2 with the Hill orthotropic yield criterion][ 44 ]. It uses a tangent stiffness algorithm [ 29 ], [ 52 ] for the update of the plastic stress. This algorithm consists of the calculation of a new elastic-plastic stiffness matrix every time step. The procedure is to a certain extent described in [ 29 ], but in the context of an implicit FE algorithm. However the main equations that are used in this implementation do correspond to the equations in [ 29 ]. An important

drawback of this stress update algorithm is that the solution may drift away from the yield surface. The implementation of a return mapping algorithm [ 7 ] would avoid this problem.

The routines that deal with material model 33 are `solde`, `matin`, `initlz`, `f3dm33`, `sets33` and `inse33`. A general description of the structure and data handling of DYNA3D can be found in [ 17 ]. Subroutine `matin` reads in the material data from the input file. Inside this routine there is a call to subroutine `sets33` which reads in all the material specific data for material model 33. In subroutine `initlz` calls are made to subroutines that initialise values that are required during the solution phase but that are not separately defined in the input file. The routine that initialises the values related to material model 33 is the `inse33` subroutine. It calculates the  $q$  matrix, which is the transformation matrix from element to material axes. The calculation depends on the value of `AOPT`. Depending on the material model used different subroutines are called in the `solde` subroutine that processes all the solid elements (a different routine is used for beams and shells). In this example, for material model 33, five routines are called `hvpac1`, `rstrss`, `f3dm33`, `bulkq`, `engbrk`, `hvpac2`. The `hvpac1` and `hvpac2` routines pack and unpack the data for the material model. Subroutine `hvpac1` loads the element stress tensor and all the necessary auxiliary variables. Subroutine `rstrss` calculates the rotated stress tensor. Subroutine `f3dm33` is the main routine for each material model and calculates the new stress tensor, the element internal energy. Subroutine `bulkq` calculates the critical timestep and bulk viscosity. Subroutine `engbrk` calculates the kinetic energy for every element. Subroutine `hvpac2` stores the new stress tensor and the auxiliary variables back in the main database. More detailed information on material 33 is provided in APPENDIX C.

#### 5.4.2 Implementation of the New Material Model

In order to implement all of the model features discussed in the previous sections the anisotropic model 33 was taken as a starting point. The improvements require changes to several routines in the code, not just the `f3dm33` routine. The improved model requires many more input parameters. In summary the additions to model 33 are:

- ▶ Strain rate dependent yield stress and hardening modulus.
- ▶ Combined isotropic/kinematic hardening.
- ▶ Equation of state.
- ▶ Spall models.

Because of the numerous changes required to the code it was decided to create two new material models, one with an EOS and one without. In the future this will allow for different improvements to be incorporated in only one of the new models, as the models are likely to be used to simulate rather different events. If only one model was available this would lead to a very complex model with a multitude of input parameters. Due to its complexity, and hence lengthy coding, it would also become very computationally intensive. The two new models were assigned numbers 52 and 53, the first being the model without EOS, the latter with an EOS.

Subroutine Name	Purpose of Subroutine	Changes Made
bczdtb	Constitutive evaluation for bciz triangular shell elements	A call to subroutine shl52s was added
blkdat	All data blocks are defined in this routine	The data block nconst, which defines the number of auxiliary variables for every material model was modified so that the new models have 5 auxiliary variables alike to model 33.
bulkmd	Calculates or gathers the bulk modulus for every material	Calculation added for materials 52 and 53
c0tdtb	Evaluates constitutive routines for Cst c0t shell elements	Calls to routines shl52s and shl53s were added.
cnmdtb	Evaluates constitutive routines for the blytsy and berwsh elements	Calls to routines shl52s and shl53s were added.
conmd	Evaluates constitutive routines for thick shells	Calls to routines shl52s and shl53s were added.
conmds	Evaluates constitutive routines for Hughes-Liu shells	Calls to routines shl52s and shl53s were added.
in3dis	Initialisation routine for shell elements	Calls to routines insh52 and insh53 were added.
initlz	Main initialisation routine	Calls to routines inse52 and inse53 were added. These are the routines that initialise solid elements.
matin	Reads in the material cards from the input file.	Extra lines were added that read in the material cards for materials 52 and 53.
printm	Writes the material data that was read in matin to the d3hsp file.	Extra lines were added that output the data for material 52 and 53.
shearm	Calculates the shear modulus	Extra lines were added to do this for material 52 and 53.
shlmat	Evaluates the constitutive routines in the shell material driver	Calls to routines shl52s and shl53s were added.
solde	Evaluates the constitutive routines for solid elements in the solution phase	Calls to routines f3dm52 and f3dm53 were added.
soldrv	Evaluates the constitutive routines for solid elements in the material driver	Calls to routines f3dm52 and f3dm53 were added.

Table 5.1: Overview of modified DYNA3D subroutines.

As a first step the new material models where set up as copies of model 33. In order to do this all routines uniquely related to model 33 where duplicated and renamed to reflect the fact that they where related to models 52 and 53. The routines that where duplicated are: f3dm33, inse33, sets33, shl33s, insh33. The duplicated routines where then named f3dm52 and f3dm53, inse52 etc. Further changes where required in 19 routines. These changes are summarised in Table 5.1.

### 5.4.3 Validation of Initial Anisotropic Model

In order to validate the implemented routine a series of tests was conducted using the material driver in LLNL-DYNA3D. This material driver allows for a strain history to be specified and the material model response to this history is then calculated.

First of all material parameters where used such that the yield surface is equivalent to a Von Mises yield surface. This allowed the results to be compared to the results obtained with the isotropic-kinematic hardening model using Von Mises plasticity (Material 3) [ 61 ]. The material parameters that where used are summarised in Table 5.1. The coefficients of the yield function are according to Equations 9 and 10.

A strain history was applied in the X-direction. This consisted of the stretching of the material in the X-direction, until a strain of 1.5% was reached. Then the material was compressed to a compressive strain of -1.5%. Finally the material was again stretched until a strain of 1.5% was reached. This deformation history results in a typical hysteresis loop observed in cyclic loading tests. As can be seen from Figure 5.4 the results obtained with the anisotropic model and model 3 are identical and are for the case of pure isotropic hardening ( $\beta = 1$ ).

Parameter	Value
Young’s Modulus, E	210 GPa
Poisson’s Ratio, $\nu$	0.3
Yield Stress, $\sigma_Y$	600 MPa
Plastic Modulus, $E_p$	1GPa
Hardening Parameter $\beta$	0 or 1

Table 5.2: Material Model 3 Constants for Steel.

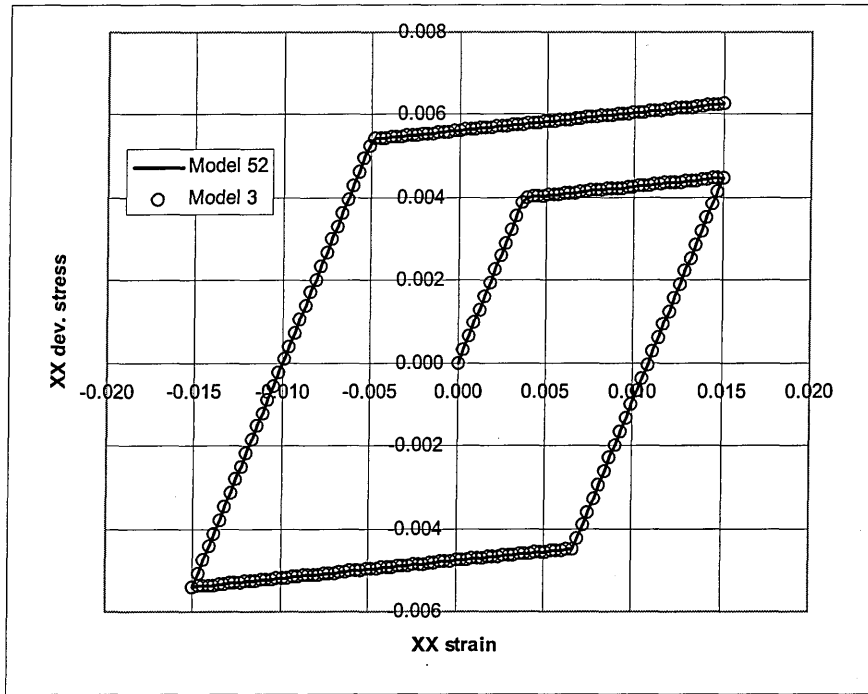


Figure 5.4: Anisotropic vs Model 3 Comparison: Isotropic Hardening.

#### 5.4.4 Implementation of Kinematic Hardening

The extension of the model to allow for combined isotropic-kinematic hardening requires one extra material parameter,  $\beta$ . This parameter is provided as one of the entries in the extra material data `cm2(*)`. The main modifications that are required to the `f3dm52` and `f3dm53` subroutines are due to the fact that one has to now use  $\xi$  instead of  $\sigma$ . Recall that  $\xi$  is defined as:

$$\xi = \sigma - \alpha.$$

This requires storing  $\alpha$  as a history variable and the inclusion of an extra term in the calculation of  $C_{ep}$ .

The 6 components of the backstress tensor  $\alpha$  are stored in common `/aux14/`. This ensures that they are permanently stored and can be updated every timestep. The extra memory to store these variables in common `/aux14/` is available because the parameter `nconst` was increased to 64 in the blockdata `blkdat`.  $C_k$  is also stored in common `/aux14/`.

As part of the initialisation of variables in the `f3dm52` and `f3dm53` subroutines,  $C_k$  is calculated and the current yield stress is calculated using

$$\sigma_Y = \sigma_{Y_0} + \beta H' \bar{\epsilon}^p$$

where  $\sigma_{Y_0}$  is depending on the value of the current strain rate. The next modifications are in the calculation of the equivalent stress. There  $\xi$  has to be used instead of  $\sigma$ .



$$\bar{\sigma} = \sqrt{\frac{3}{2} \frac{1}{F+G+H} \left( F(\xi_{yy} - \xi_{zz})^2 + G(\xi_{zz} - \xi_{xx})^2 + H(\xi_{xx} - \xi_{yy})^2 + 2L\xi_{zy} + 2M\xi_{zx} + 2N\xi_{xy} \right)}$$

The following modification is due to the extra term

$$(1-\beta)C_k \frac{\partial f}{\partial \sigma} : \frac{\partial f}{\partial \sigma} \quad (\text{Eq. 5.64})$$

that appears in the plastic multiplier  $\lambda$  when kinematic hardening is considered. The general expression for  $\lambda$ :

$$\dot{\lambda} = \frac{\frac{\partial f}{\partial \sigma} : (C : \dot{\epsilon})}{\frac{\partial f}{\partial \sigma} : (C : \frac{\partial f}{\partial \sigma}) + (1-\beta)C_k \frac{\partial f}{\partial \sigma} : \frac{\partial f}{\partial \sigma} + \sqrt{\frac{2}{3}}\beta C_i} \quad (\text{Eq. 5.65})$$

This extra term also shows up in the calculation of the tangent stiffness matrix  $C_{ep}$ ,

$$C_{ep} = C - C : \frac{\frac{\partial f}{\partial \sigma} : C}{\frac{\partial f}{\partial \sigma} : (C : \frac{\partial f}{\partial \sigma}) + (1-\beta)C_k \frac{\partial f}{\partial \sigma} : \frac{\partial f}{\partial \sigma} + \beta \sqrt{\frac{3}{2}} C_i}.$$

$C_{ep}$  is calculated by a call to a separate subroutine `cepx52` that calculates the tangent stiffness matrix using the above formula. In order to achieve this the values of  $\xi_{ij}$ ,  $\beta$  and  $C_k$  are passed to this routine, and the extra term Eq. 5.64 is calculated in this new subroutine and incorporated in the denominator of  $C_{ep}$ . In the calculation of the plastic strain increment tensor one has to again use  $\xi_{ij}$  instead of the stress tensor.

The final modification is the update of the backstress tensor  $\alpha_{ij}$  which is performed at the end of the subroutine, before the stresses are rotated back to their current configuration. One simply calculates the increment in backstress from the following formula:

$$d\alpha = (1-\beta)C_k d\epsilon^p$$

#### 5.4.5 Validation of Combined Isotropic-Kinematic Hardening Model

In order to validate the implementation of the kinematic hardening a simulation was performed using the material driver. The case that was analysed was identical to the one presented in section 5.4.3. The same material properties were used and the material was subject to the same strain history. The only difference was the value of  $\beta$  was set to

zero, ie. pure kinematic hardening. The results are shown in Fig. 10 and are identical to the ones obtained with material model 3.

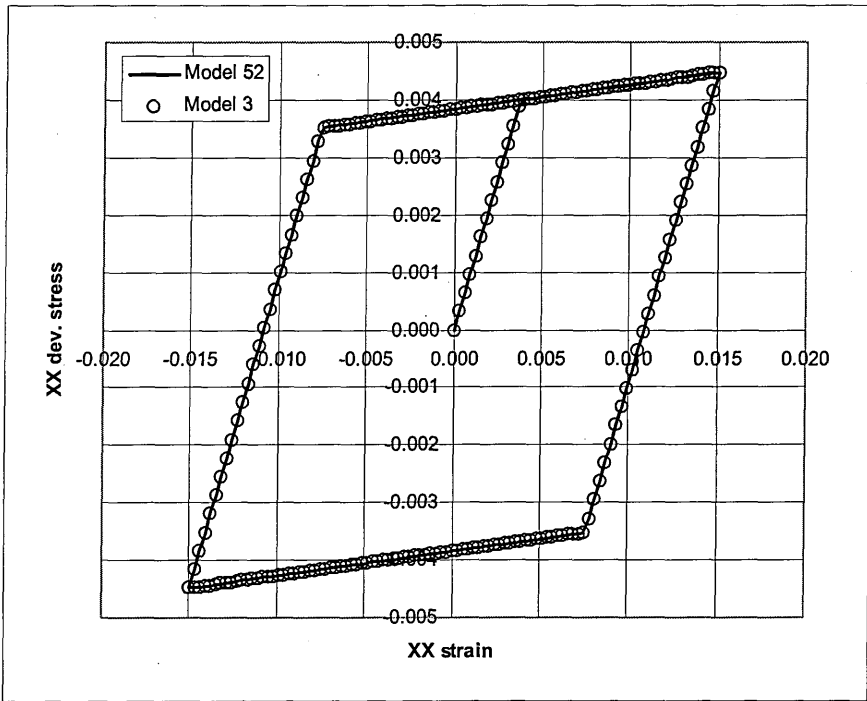


Figure 5.5: Anisotropic vs Model 3 Comparison: Kinematic Hardening.

### 5.4.6 Implementation of Strain Rate Dependency

Due to the fact that several new input parameters will be required for the new model it was decided to add six extra material cards to the input format for the models 52 and 53. This will enable not only the definition of the new material parameters that are required for the improvements described below, but this will also leave sufficient free space for any future developments to these material models. In order to add extra material cards there are three variables that are of importance:

```
mmauxs
lthrpr
lc2
```

lc2 is the pointer that points to the beginning of where auxiliary material data is stored in the main database a. Initially the space starting from lc2 is initialised in dynai (line 614) using:

$$lc3=lc2+966*2*nmnat$$

This means that per material there is space for 2x966 numbers, and this data is stored between lc2 and lc3. In the routine matin, which reads in the material data, the value of mmauxs is incremented by 1 every time a set of extra material data is read. Consequently at the end of the matin subroutine mmauxs is equal to the number of blocks

of auxiliary material data that has been read in and have to be stored in  $a(lc2, \dots, lc3)$ . The value of  $lc3$  is then finally calculated as:

$$lc3 = lc2 + 966 * mmauxs$$

This means that there are 966 memory addresses available per material. The value of  $lthrpr$  is the pointer that points to the first value of the auxiliary material data for every material. This pointer is stored in the array with 'primary' material data, usually called  $cm(*)$  throughout the program. This pointer is stored in  $cm(48, n)$ , where  $n$  is the material number. This operation is performed in subroutine *matin*.

In order to read in the extra material data for the new material models, the subroutines *sets52* and *sets53* have to be moved to allow for the reading of shell and beam section properties prior to this. This is also the case for other materials where extra material data is required such as models 22, 23 and 41. In subroutine *sets52* and *sets53* lines are required, similarly to for example *sets22*, to read in the 6 extra material cards which are stored in an array called  $cm2(*)$ . This array is passed to the main database *a* in the call statement to *sets52* and *sets3* in subroutine *matin* by using the pointer  $lthrpr$ .

In order to make available the material data to the constitutive routines *f3dm52* and *f3dm53* again the pointer  $lthrpr$  is used. Immediately before the calls to *f3dm52* and *f3dm53* the value of the pointer  $lthrpr$  is read from  $cm(48, n)$ , and value  $a(lthrpr)$  is added as an argument in the calls to *f3dm52* and *f3dm53*. These modifications are required in both subroutines *solde* and *soldrv*. Furthermore, the main database *a* has to be made available in these subroutines which is achieved by adding the common block */amain/* an pointer *ipblank*:

```
common /amain/ipblank  
pointer(ipblank,a(1))
```

In the dummy variable list of routines *f3dm52* and *f3dm53*  $a(lthrpr)$  is called *cm2*, which means that the extra material data will be available in these routines under this name.

The extra cards available for material data can be used to store strain rate dependency related material properties. In order to allow different strain rate models to be used within this material model it was decided to implement a tabulated strain rate dependent model. This has the further advantage that the implementation can be validated against the existing tabulated strain rate dependent model 19. This material model uses Von Mises plasticity, and allows for strain rate dependency of the Young's modulus, the yield stress, the hardening modulus and failure strain. The extra data that is required in the material cards is four values referring to a load curve number for each of the strain rate dependent parameters. The values of each strain rate dependent parameter are entered as a curve of this material parameter versus the strain rate. In order for the load curve data to be read in, stored and initialised correctly several modifications have to be made. These modifications are summarised in Table 5.3.

Subroutine Name	Purpose of Subroutine	Changes Made
bczdtb	Constitutive evaluation for bciz triangular shell elements	Related to shell elements – no change has been made
bulkmd	Calculates or gathers the bulk modulus for every material	Same for material 19 and 52/53 – no change has been made
c0tdtb	Evaluates constitutive routines for Cst c0t shell elements	Related to shell elements – no change has been made
cnmdtb	Evaluates constitutive routines for the blytsy and berwsh elem	Related to shell elements – no change has been made
conmds	Evaluates constitutive routines for Hughes-Liu shells	Related to shell elements – no change has been made.
elem2d	main subroutine for calling two-dimensional elements	A change was made so that the creation of a scale factor array, used to calculate the minimum shell element timestep is also performed for materials 52 and 53
hghliu	main subroutine for hughes-liu shell	A change was made so that the routine hldiag is also called for materials 52 and 53. This routine calculates the minimum element dimensions, both diagonal and side, and the element area. These values are stored in common /sounds/.
in3dis	Initialisation routine for shell elements	Calls to routines insh52 and insh53 were already added.
initlz	Main initialisation routine	Calls to routines inse52 and inse53 were already added. These are the routines that initialise solid elements.
matin	Reads in the material cards from the input file.	Extra lines were already added that read in the material cards for materials 52 and 53.
matinc	Checks load curve data for materials.	Statements are added that check the four load curves that are required for materials 52 and 53.
printm	Writes the material data that was read in matin to the d3hsp file.	Extra lines were added that output the data for material 52 and 53.
shearm	Calculates the shear modulus	Extra lines were already added to do this for material 52 and 53.
shlmat	Evaluates the constitutive routines in the shell material driver	Calls to routines shl52s and shl53s were already added.

solde	Evaluates the constitutive routines for solid elements in the solution phase	Calls to routines f3dm52 and f3dm53 were already added.
soldrv	Evaluates the constitutive routines for solid elements in the material driver	Calls to routines f3dm52 and f3dm53 were already added.

Table 5.3: Overview of DYNA3D Routines Modified for Strain Rate Dependency .

These changes make available the load curve numbers that define the strain rate dependency of the material parameters, but to access the load curves themselves two more arguments need to be added in the call statements to f3dm52 and f3dm53. These two variables are called npc and plc. The variable npc is an array with pointers to the specific load curve that is required, while plc is the array containing the actual tabulated data of all load curves.

The modifications that are required to subroutines f3dm52 and f3dm53 will be described in this paragraph. First of all lines have to be added that read the load curve numbers from the cm2(\*) array, these values are read into i1, ..., l1. Next for those four load cures the pointers to the actual data in plc are read from npc(\*) and stored in i2, ..., l2. In order to interpolate the yield stress and hardening modulus from the load curves one has to calculate the current effective strain rate:

$$\begin{aligned} d\dot{\epsilon} = & \sqrt{F \left( \frac{Gd\dot{\epsilon}_y - Hd\dot{\epsilon}_z}{FG + GH + HF} \right)^2 + G \left( \frac{Fd\dot{\epsilon}_x - Hd\dot{\epsilon}_z}{FG + GH + HF} \right)^2} \\ & + H \left( \frac{Fd\dot{\epsilon}_x - Gd\dot{\epsilon}_y}{FG + GH + HF} \right)^2 + \frac{d\dot{\epsilon}_{yz}^2}{2L} + \frac{d\dot{\epsilon}_{zx}^2}{2M} + \frac{d\dot{\epsilon}_{xy}^2}{2N} \end{aligned}$$

This value of the effective strain rate is used to interpolate the correct values for Young’s modulus, yield stress, hardening modulus and failure strain, this is performed in subroutine gtp19.

The final change that is required due to the fact that the routine is vectorised. The yield stress, hardening modulus and failure strain can now be different for every element in the vector that is being processed. However the variables are initialised as single values in f3dm33. In subroutines f3dm52 and f3dm53 these variables have been converted into array of size lnv. lnv is the variables that contains the size of the current vector that is being processed. This requires various minor changes throughout the routine. For example the intermediate variable ttt has to be converted to an array of size lnv as well.

### 5.4.7 Validation of Strain Rate Dependency

In order to validate the implementation of the strain rate dependent yield stress and hardening modulus the results of the new material model can be compared to the ones obtained using material model 19. The material properties and applied strain history is

identical to section 5.4.3. Material model 19 is a rate dependent isotropic hardening model, and uses a Von Mises yield criterion [ 61 ]. The results for isotropic hardening ( $\beta = 1$ ) are shown in Figure 5.6, and show identical results to model 19. Material model 19 only allows for isotropic hardening, so a direct comparison with this model is impossible. However from Figure 5.7 it can be seen that the kinematic hardening effect ( $\beta = 0$ ) is taken into account correctly.

Parameter	Value
Young's Modulus, E	210 GPa
Poisson's Ratio, $\nu$	0.3
Yield Stress, $\sigma_Y$	600 MPa
Plastic Modulus, $E_p$	1GPa
Hardening Parameter $\beta$	0 or 1

Table 5.4: Material Parameters for Steel.

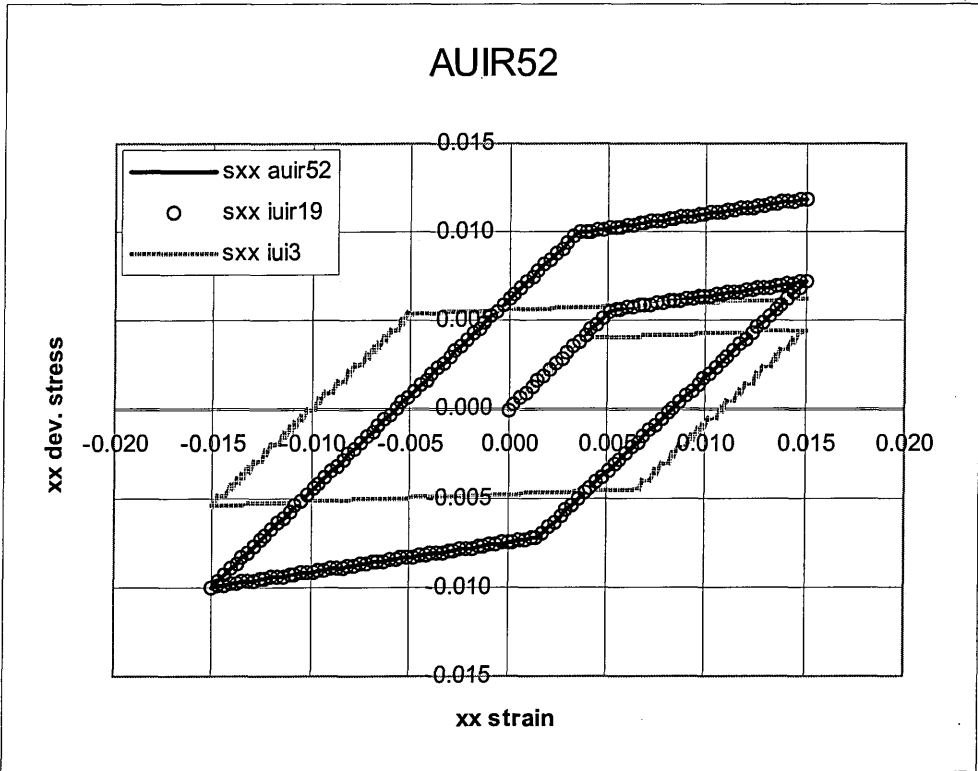


Figure 5.6: Anisotropic vs Model 19 Comparison: Isotropic Hardening.

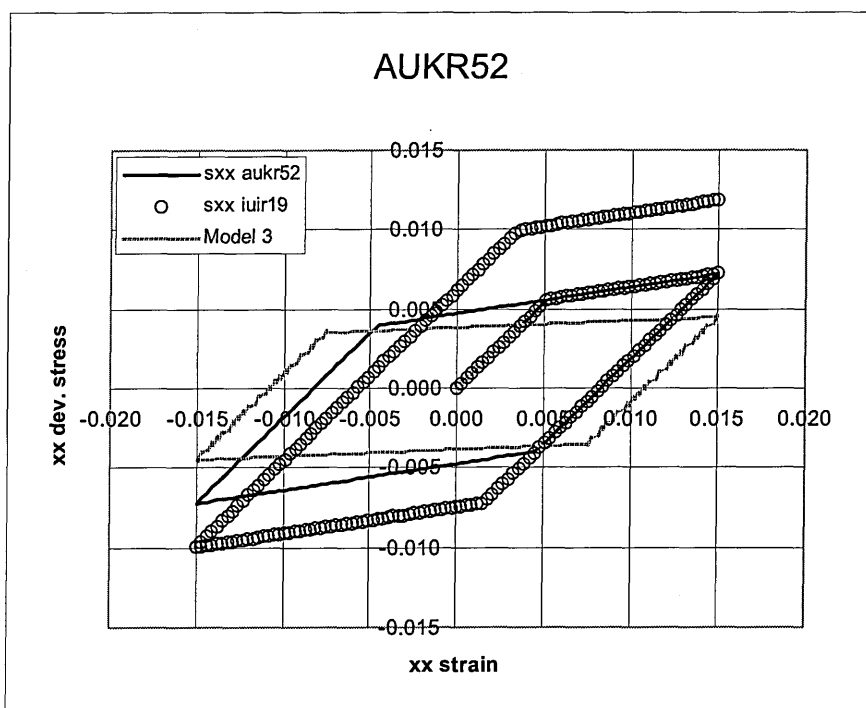


Figure 5.7: Anisotropic vs Model 19 Comparison: Kinematic Hardening.

The addition of kinematic hardening and strain rate dependency concludes the development of material model 52.

#### 5.4.8 Implementation of Equation of State

In order for the anisotropic material model to be used in high velocity impact simulations it is necessary to implement an equation of state. The main difference with material models that do not use an EOS is that the pressure component is not dealt with in the f3dm part of the constitutive model, but in a separate set of subroutines. This implies that the f3dm part of the constitutive model should only deal with the deviatoric part of the constitutive relation. In f3dm52 the spherical part of the stress tensor is updated using the elastic constitutive relation, and several other intermediate variables are calculated using the full stress tensor. In f3dm53 several modifications were made to decouple the spherical part of the stress tensor and the deviatoric part.

First of all some changes were required to initialise the equation of state data. The reading in of the EOS data line is automatic as long as a valid EOS number is specified on the first material data card. The number of arguments passed to the inse53 routine was increased to pass all EOS related data. In inse53 this EOS related data is initialised and stored in the auxiliary variables and stored in common /aux14/ after the stress tensor. This ensures that this data is available to other subroutines when required.

The main changes involved the calls to f3dm53, and in f3dm53. Without an equation of state calls are made to f3dm52, bulkq and engbrk after the rotation of the stresses in rstrss. The purpose of bulkq and engbrk is to calculate the new critical timestep and bulk

viscosity, and the momentum and kinetic energy respectively. If an equation of state is used three extra routines are called:

sueos  
hieupd  
eqos.

The sueos subroutine has for purpose to calculate the element speed of sound. The hieupd subroutine calculates the intermediate value of the internal energy, and eqos calculates the new pressure and internal energy. The procedure used to update pressure and internal energy will not be discussed here but can be found in [ 17 ], [ 41 ].

All remaining modifications were in f3dm53 and are necessary to ensure that all required data is available to sueos, bulkq, hieupd and eqos. Subroutine hieupd requires the old stress tensor. In order to make the old stress tensor available to hieupd the stress components are written to sigv1(\*), ..., sigv6(\*), and stored in the second to seventh entry in common /aux11/. They are read from this position in hieupd. The other modifications are related to the update of the stress tensor. This update should only be related to the deviatoric part of the stress tensor, and therefore, instead of adding the full elastic increment to the old stress, only the deviatoric part is added. The addition of the plastic part of the stress increment does not require any changes as plastic deformation is incompressible and hence only causes a change in deviatoric stress. A final change is required at the end of the routine. If there is no plasticity the code jumps straight to the end of the routine, skipping the plasticity part. However, as no change is made to the stress tensor this tensor will still be the full stress tensor. Therefore this tensor is overwritten by its deviatoric equivalent at the end of the routine.

#### 5.4.9 Validation of the Implementation of the Equation of State

The implementation of the EOS was validated by calculating the curve relating pressure to impact speed of aluminium 2024 from a series of DYNA3D analyses, and comparing it to experimentally data from Kinslow [ 51 ]. This curve was chosen as the check as it is dependent only on the equation of state, and can be calculated from a series of DYNA3D analyses. The simplest experiment consists of the impact two flat plates of identical material. If one is stationary then the particle velocity is one half the impact velocity.

The mesh that was used to simulate this plate impact is shown in Figure 5.8. As it is a 1D problem four symmetry planes are specified to ensure no release waves are generated at the sides of the plate. Instead of modelling another plate with equal and opposite velocities the plate impacts a rigid wall, which has the same effect. In this case the particle velocity will be equal to the impact velocity. Time histories were taken from nodes on the centreline of the mesh. A time history is shown in Figure 5.9. The pressure levels in the material can be read from this time history for every impact velocity. These points can then be plotted in a curve and compared to the experimental data. The result of this comparison for three different velocities is shown in Figure 5.10. The results are almost identical to the experiment. For comparison the same simulation was also performed using the elasto-plastic hydrodynamic material model in



DYNA3D, model 10 [ 61 ]. It can be seen that the results obtained with the new material model are identical to model 10.

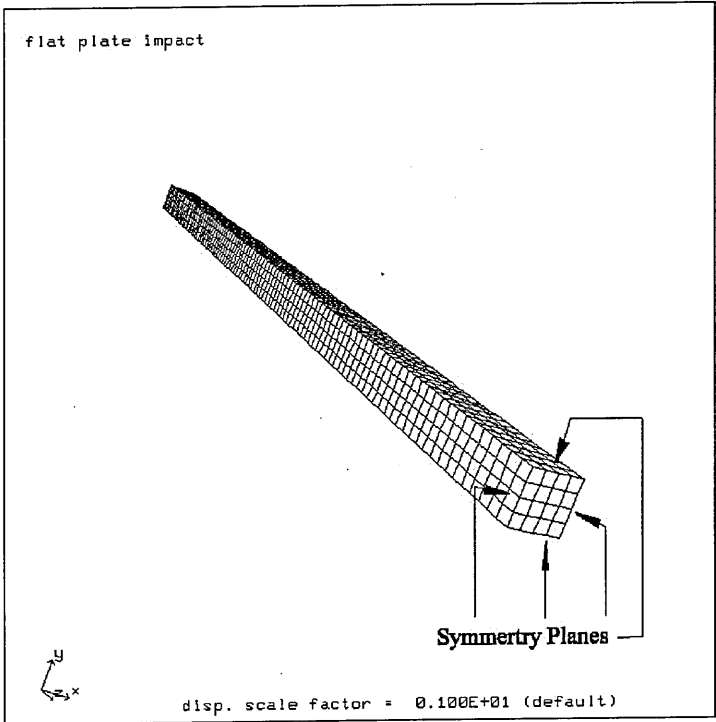


Figure 5.8: Mesh used for EOS Validation.

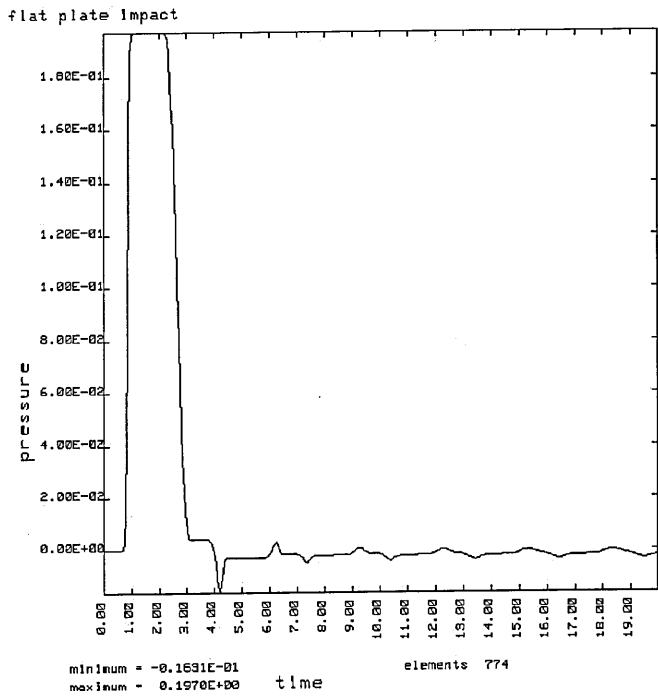


Figure 5.9: Pressure Time history during Plate Impact.

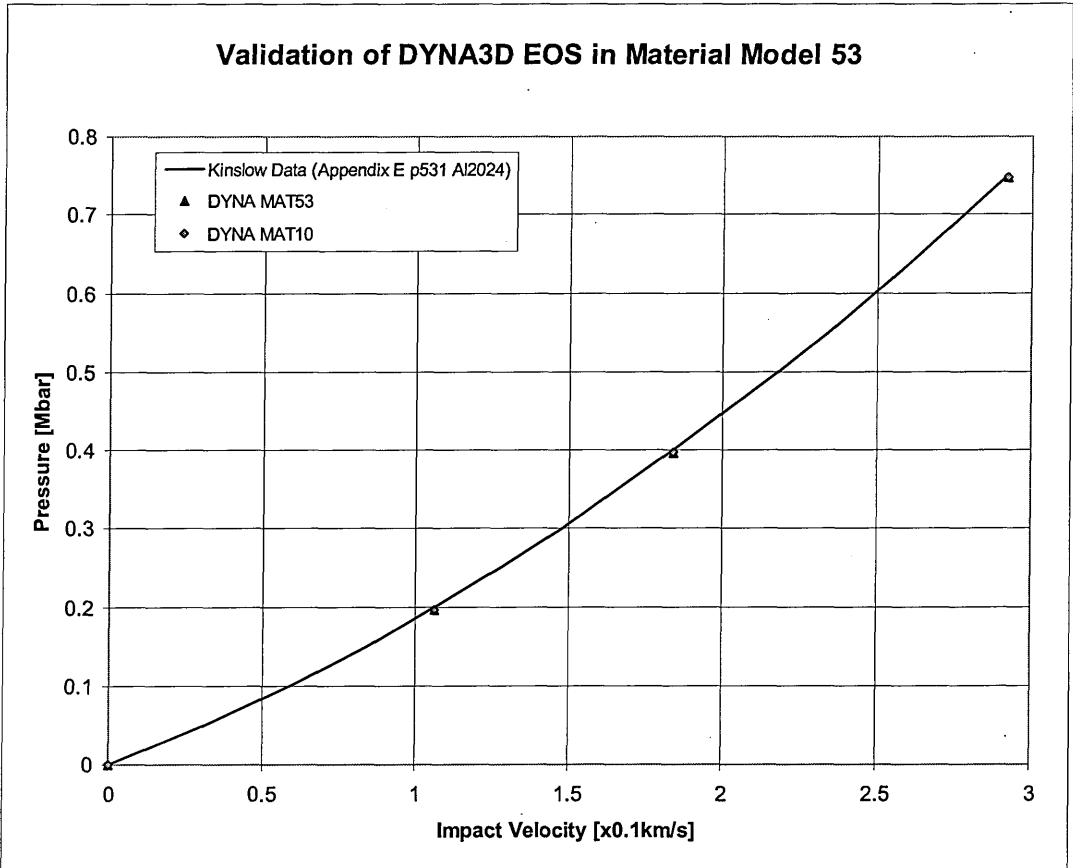


Figure 5.10: Validation of the EOS Implementation in Material Model 53.

#### 5.4.10 Implementation of Spall Models

Three spall models have been implemented in the new material model:

- ▶ A pressure cut-off model
- ▶ A maximum principal stress model
- ▶ Grady's model

In the case of a pressure cut off or the maximum principal stress criteria, the only data that is needed is the cut off value. This value is specified in the material input cards and stored in common /aux14/ in inse53. The current pressure or maximum principal stress is easily calculated in f3dm53 and the resulting value is simply compared with the user specified cutoff value. If spall is detected then the complete stress tensor is set to zero.

Grady's spall criterion requires two extra material parameters, the failure strain  $\epsilon_{fail}$  and the critical fracture toughness  $K_c$ . These were made to be read in from the material cards and stored in cm2(\*). The Eq. 5.56, Eq. 5.57 and Eq. 5.58 were implemented to detect spall. Similarly to the first two criteria the stress tensor is set to zero once spall is detected.

5.4.11 Validation of the spall models

In order to validate the spall models the material driver in DYNA3D was used. A single element was subjected to an identical strain history in X, Y and Z directions. This then creates a state of hydrostatic tension. Without a spall model the pressure decreases as the material is stretched. When the spall models are activated one should expect the pressure to be set to zero once it reaches a certain value that corresponds to the spall strength of the material. This can clearly be observed in Figure 5.11. Grady’s spall criterion is triggered earlier than the other criteria. This is simply due to the material constants that were used (Al 2024), and the arbitrary value of the pressure or stress cut-off value for the first two spall criteria. In this case the value was set to 1GPa.

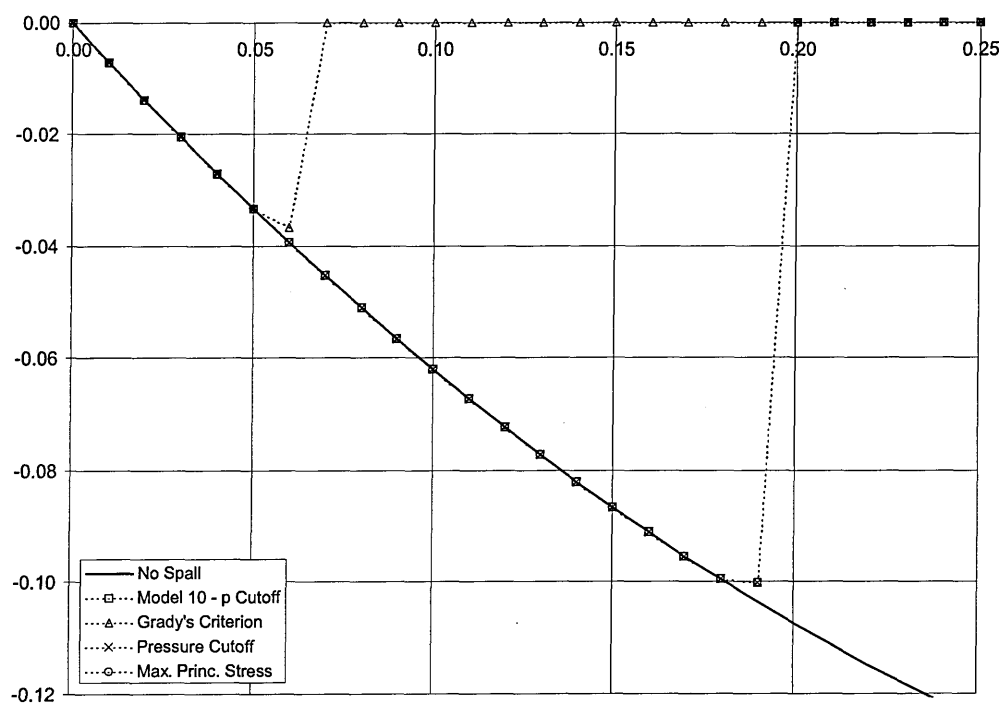


Figure 5.11: Validation of the Spall Model Implementation in Material Model 53.

5.5 Simulation of Plate Impact on the Aluminium alloy 7010 – T6

5.5.1 Introduction

The purpose of this section is to evaluate the performance of the material model for the more demanding problem of a plate impact on an alloy with anisotropic material properties. The effect of orientation on the mechanical properties of metals and alloys is well known and has been studied extensively under quasi-static conditions. This can occur at the unit cell level, due to preferred orientations in the grain structure itself, or due to the distribution of secondary phases present within the microstructure. A more complete discussion of such behaviour has been made by Smallman [ 84 ]. In contrast, similar measurements made at dynamic strain-rates are nowhere near as extensive. Gray *et al.*[ 34 ] investigated a cold-rolled and annealed zirconium. Quasi-static testing showed that peak stresses were ca. 2.5 times greater in the through-thickness direction

when compared to the in-plane direction for the same plate. During shock loading, results showed that the variation of the Hugoniot Elastic Limit (HEL) were consistent with the quasi-static measurements. Orientation was also shown to have a significant effect upon damage evolution, but a minimal effect on the pull-back (spall) signal seen in recorded VISAR traces. Similar measurements have also been made on a eutectoid 1080 rail steel [ 35 ]. Here the material is crystallographically isotropic, but possesses microstructural anisotropy due to the presence of manganese sulphide (MnS) stringers that orientate themselves along the rolling direction. The behaviour of aluminium alloys under shock loading has been studied in some depth [ 72 ], [ 90 ]. This is due to their low densities and (in some alloys at least) high strengths which has encouraged their use as light-weight armours and airframes. Possibly the most thorough study on a single alloy was made by Rosenberg *et al.* [ 78 ]. Here, they showed that in the alloy 2024 (Al+Cu+Mg), the HEL and spall strength followed the same trends as the quasi-statically measured yield strength, with the lowest measured in the solution-treated material. They also demonstrated that orientation also had an effect upon the spall strength. Here the lower spall strength was measured perpendicular to the rolling direction when compared to that measured parallel to it. In this paragraph simulation results from the alloy 7010-T6, a high-strength airframe alloy (Al+Zn+Mg+Cu) that show similar behaviour are used to demonstrate the capabilities of the new material model. The recorded stress traces, Hugoniot Elastic Limit (HEL) and spall strength in the short transverse and longitudinal (rolling) directions are compared with experimental results.

### 5.5.2 Experimental Setup

The material data that is available is summarised in Table 5.5.

Orientation	$c_L$ (mm us <sup>-1</sup> )	$c_s$ (mmus <sup>-1</sup> )	$\rho$ (g cm <sup>-3</sup> )
Longitudinal	6.24±0.03	3.06±0.03	2.81±0.03
Short Transverse	6.26±0.03	3.07±0.03	2.81±0.03
6082-T6	6.40±0.03	3.15±0.03	2.70±0.01

Table 5.5: Materials Data for Al 7010-T6 and Al 6082-T6.

From the longitudinal and shear sound speeds one can determine the elastic material constants:

$$c_L = \sqrt{\frac{K + \frac{4}{3}G}{\rho}} \tag{Eq. 5.66}$$

$$c_s = \sqrt{\frac{G}{\rho}} \tag{Eq. 5.67}$$

From these equation one obtains the shear and bulk moduli. These moduli can then be used to calculate the Young's moduli and Poison's ratios using the following equations:

$$G = \frac{E}{2(1+\nu)} \quad (\text{Eq. 5.68})$$

$$K = \frac{E}{3(1-2\nu)} \quad (\text{Eq. 5.69})$$

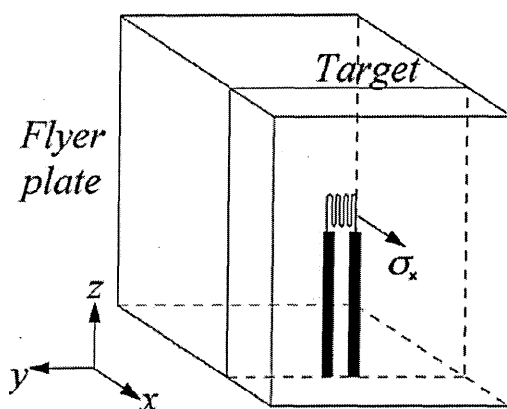


Figure 5.12: Test Configuration.

The experiments were performed on 75 mm and 50 mm bore single stage gas guns. 50 x 50 x 5 mm<sup>3</sup> tiles of the aluminium alloy 7010-T6 were cut from a single hot rolled block, and ground such that they were flat and parallel to within 5 optical fringes over 50 mm. The hot working process gives a characteristic pan-cake grain structure (with the long axis of the grains in the rolling direction), which shows dynamic recovery but not dynamic recrystallisation. Manganin stress gauges (MicroMeasurements LM-SS-125CH-048) were supported on the back of the alloy targets with 12 mm blocks of polymethylmethacrylate (PMMA). Voltage-time data from the gauges was converted to stress-time by using the calibration studies of Rosenberg *et al.* [ 79 ]. A 2.5 mm dural (aluminium alloy 6082-T6) flyer was chosen as an impactor since it had a close similarity in acoustic properties to the target. The reflected complete releases from target and flyer would interact in the centre of the 7010 target plate. Impact velocities were chosen to be around twice and three times the HEL of the material, that is ca. 450 m s<sup>-1</sup> and 895 m s<sup>-1</sup>, a test at the lower velocity of 234 m/s<sup>-1</sup> was also performed. The velocities were measured via the shorting of sequentially mounted pairs of pins.

### 5.5.3 DYNA3D Model

The main purpose of the modelling studies was firstly, to investigate the capability of implemented constitutive model to accurately predict the material behaviour under high velocity impacts. Secondly, to obtain insight into some of the experimental observations and attempt to explain their physical significance. It was clear that some of these observations challenged quite long-standing views of the physics of spallation. Modelling therefore provided a potentially powerful and independent approach to investigating the issues. The main reason for this is that hydrocodes do not make any pre-determined assumptions concerning the stress-system or wave propagation

behaviour. They solve the conservation equations and use the constitutive models and equations of state to determine dynamic the material response.

The material properties of the Al 7010 alloy listed in Table 5.6. It can be seen that when tested in the longitudinal orientation to the short transverse, 7010-T6 displays higher strengths and elongations then in the short transverse direction. From this data, and the data in Table 5.5 one can calculate all the relevant material constants that describe the material behaviour in the elastic and plastic regime. The material parameters that where used in the analyses are summarised in Table 5.7

Orientation	0.2% Proof Stress (MPa)	Tensile Strength (MPa)	%Elongation
Longitudinal	564	604	17
Short Transverse	516	563	12

Table 5.6: Quasi-static mechanical properties of 7010-T6.

Parameter	Al 7010 – T6	Al 6082 – T6	PMMA
E <sub>a</sub>	70.6 GPa		
E <sub>b</sub>	71.1 GPa		
E <sub>c</sub>	70.6 GPa		
R	1		
P	0.719		
Q <sub>bc</sub>	1		
Q <sub>ab</sub>	1		
Q <sub>ca</sub>	1		
v <sub>ab</sub>	0.342		
v <sub>ca</sub>	0.342		
v <sub>bc</sub>	0.342		
σ <sub>Y</sub>	564 MPa	250 MPa	70 MPa
H'	0.13 GPa	130 MPa	300 MPa
G <sub>bc</sub>	26.48 GPA	26.8 GPa (G)	2.3 GPa (G)
G <sub>ab</sub>	26.31 GPA	26.8 GPa (G)	2.3 GPa (G)
G <sub>ca</sub>	26.48 GPA	26.8 GPa (G)	2.3 GPa (G)
β	1		
p <sub>cut</sub>	-1.3 GPa	2.5 GPa	-
ρ	2.81 g/cm <sup>3</sup>	2.7 g/cm <sup>3</sup>	1.18 g/cm <sup>3</sup>
C	5200 m/s	5240 m/s	2180 m/s
S <sub>1</sub>	1.36	1.4	2.088
S <sub>2</sub>	0.0	0.0	-1.124
S <sub>3</sub>	0.0	0.0	0.0
Γ	2.2	1.97	0.85
a	0.48	0.48	0.0

Table 5.7: Material Properties used in Analysis

#### 5.5.4 Simulation Results

The characteristic of this plate impact problem is that it can be reduced to a 1D wave propagation. Hence building an FE model is greatly simplified. In the model that was used for this simulation the three materials were modelled as rectangular bars (Figure 5.13) of 4x4 elements. Symmetry planes were applied on all the sides. This will result in a 1D wave travelling along the length of the bar. A non-reflecting boundary condition was applied at the back of the PMMA block. This ensures that no release wave travels back through the PMMA in to the AA7010-T6 block as this would unload the test specimen. The impactor is modelled with 25 elements along the axis of impact, the test specimen and the PMMA block are modelled with 75 and 100 elements along the axis of impact respectively. A contact interface was specified between the impactor and the test specimen. The stress time histories were recorded in the elements at the back of the test specimen. These mesh resolutions were sufficient to allow the resolution of all the relevant elastic and plastic waves in the target and flyer.

The three impacts were simulated with the three different spall criteria: the pressure based spall criterion, the principal stress based spall criterion, and Grady's spall criterion.

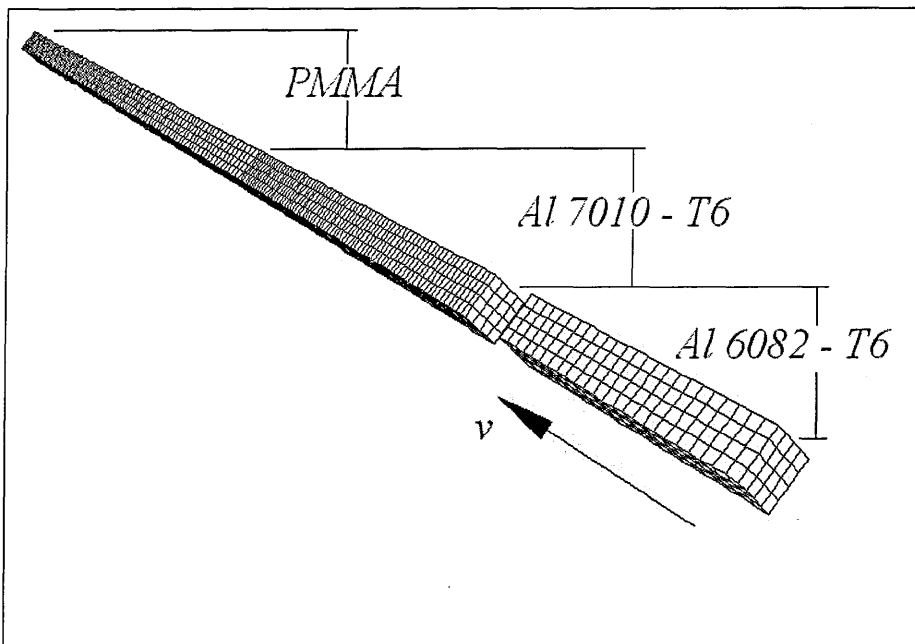


Figure 5.13: FE Mesh.

In Figure 5.14 and Figure 5.15 the simulated stress trace at the interface between the target material and the PMMA block, for the plate impact at  $234 \text{ m s}^{-1}$ , is presented. In these lower velocity impacts, it can be seen that the material does not display the characteristic reloading signals that signify the presence of hydrostatic tensile failure (spall). In Figure 5.16 and Figure 5.17, the results from the  $450 \text{ m s}^{-1}$  impact, clear spall signals can be seen. In both sets of traces a clear break in slope in the initial rising part of the trace, that is the HEL, can be seen. Furthermore, there are clear differences in the value of the HEL seen in the longitudinal and short transverse orientations. These occur at 0.43 GPa in the longitudinal orientation and 0.39 GPa in the short transverse. When

compared with the values measured in the measured experiment, 0.39 GPa and 0.33 GPa respectively, these values show reasonable correlation with the experiment. The errors are 9.3% and 15.4% respectively. It is not known what the margin of error in the measurements is. The comparison of the Hugoniot stress level is in very good agreement with the experiment. Here the simulations predict a stress of 0.63 GPa, whereas the stress recorded in the experiment is 0.65 GPa (a 3% error). Further comparison shows that the pulse width (about  $1\mu\text{s}$ ) and the reloading trace are in good agreement with the experiment as well. The main conclusion one can draw from these results is that the model as it stands is capable of simulating this type of wave propagation in anisotropic solids. The fact that different HEL's are obtained when the material is impacted in different directions demonstrates that the anisotropic plasticity model is adequate. Furthermore the good agreement of the general pulse shape and Hugoniot stress level show that the EOS is performing satisfactorily.

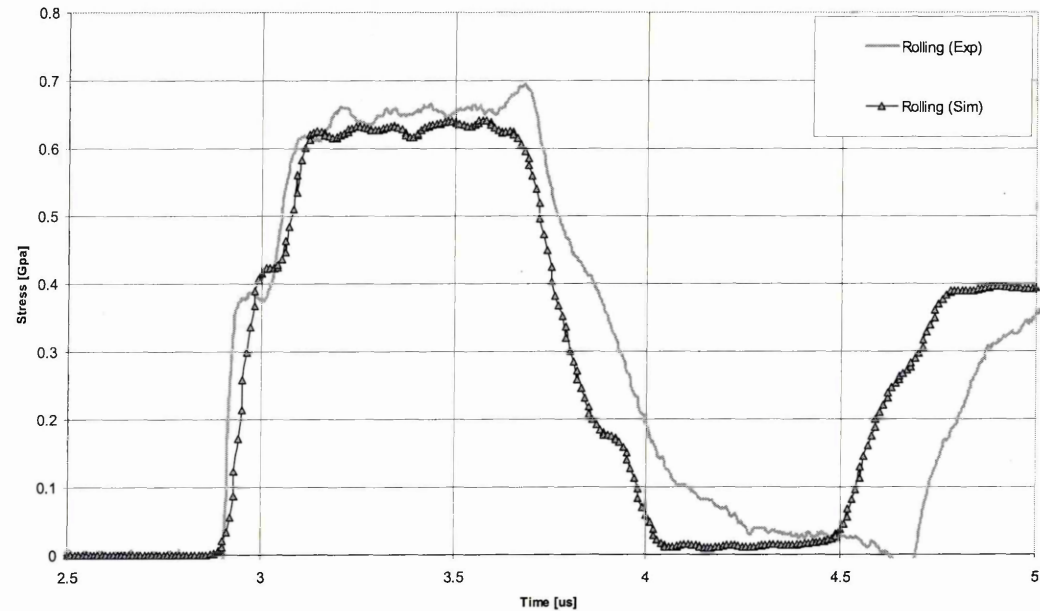


Figure 5.14: 234 m/s Impact – Rolling Direction



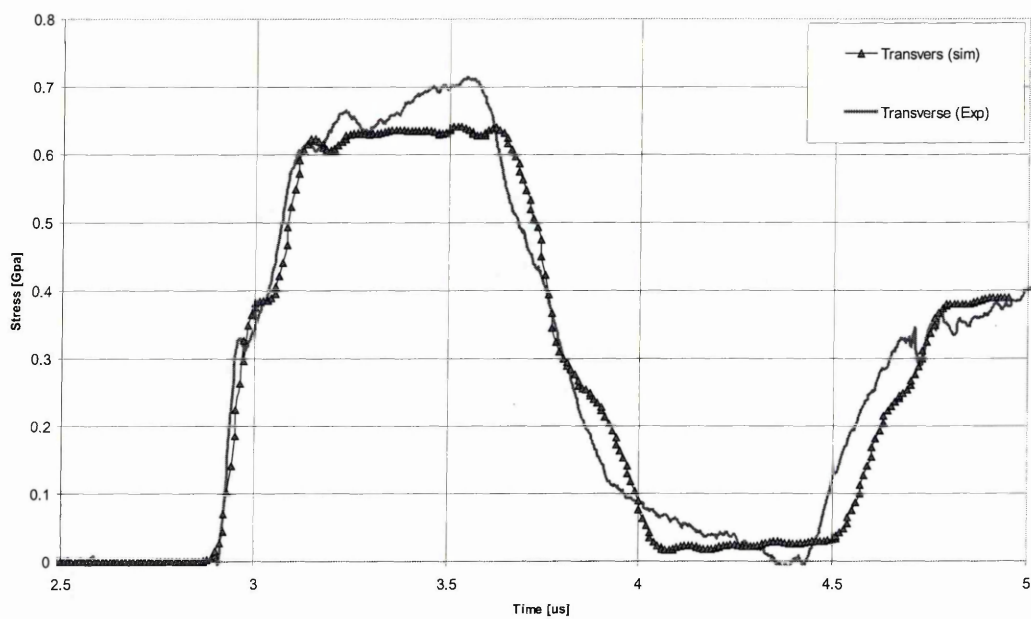


Figure 5.15: 234 m/s Impact – Transverse Direction

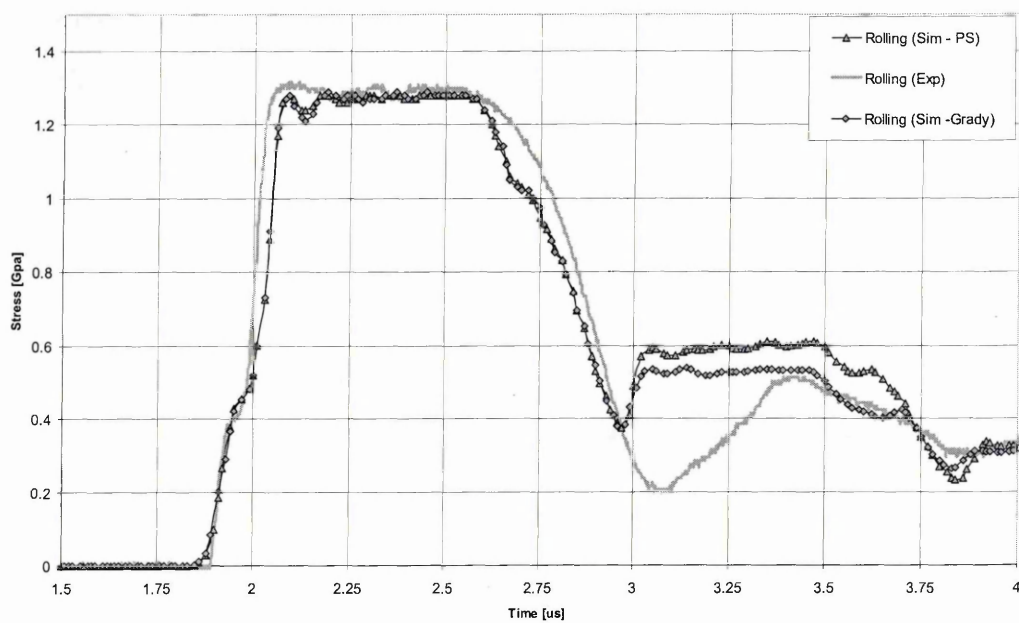


Figure 5.16: 450 m/s Impact – Rolling Direction

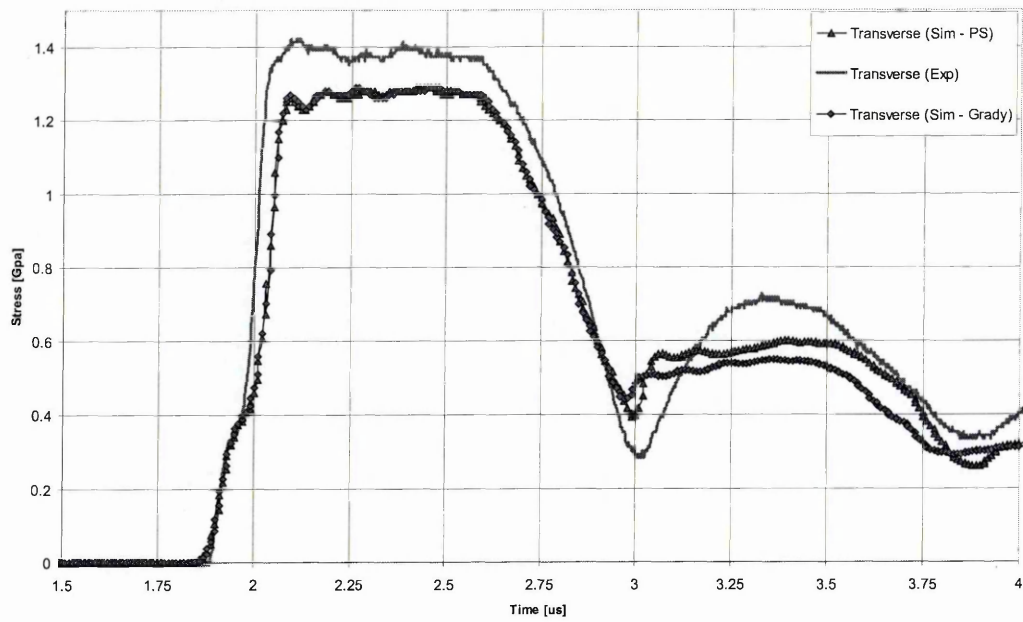


Figure 5.17: 450 m/s Impact – Transverse Direction

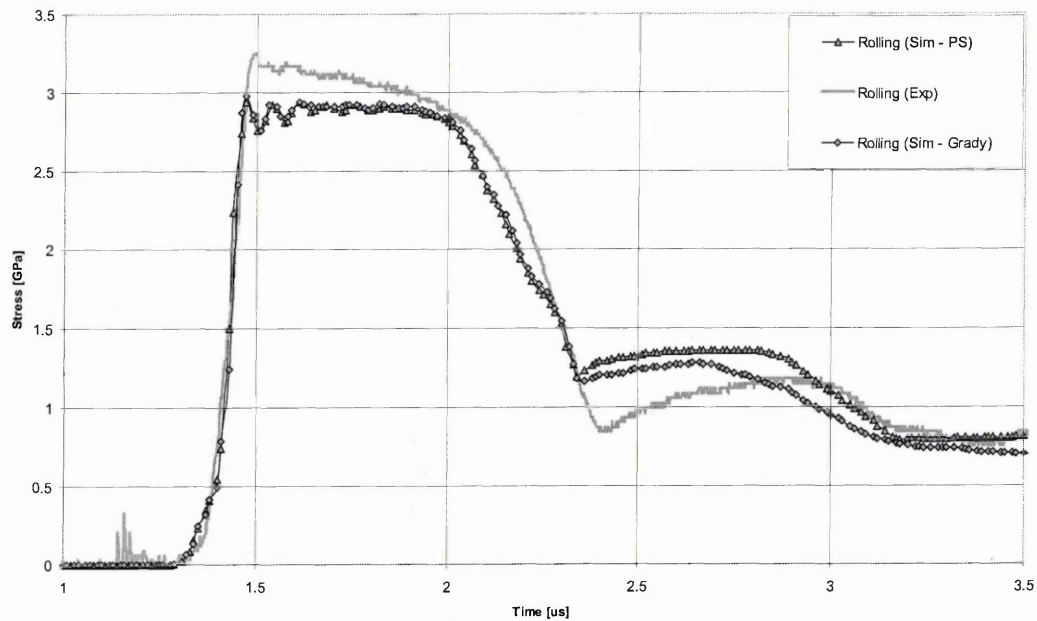


Figure 5.18: 895 m/s Impact – Rolling Direction

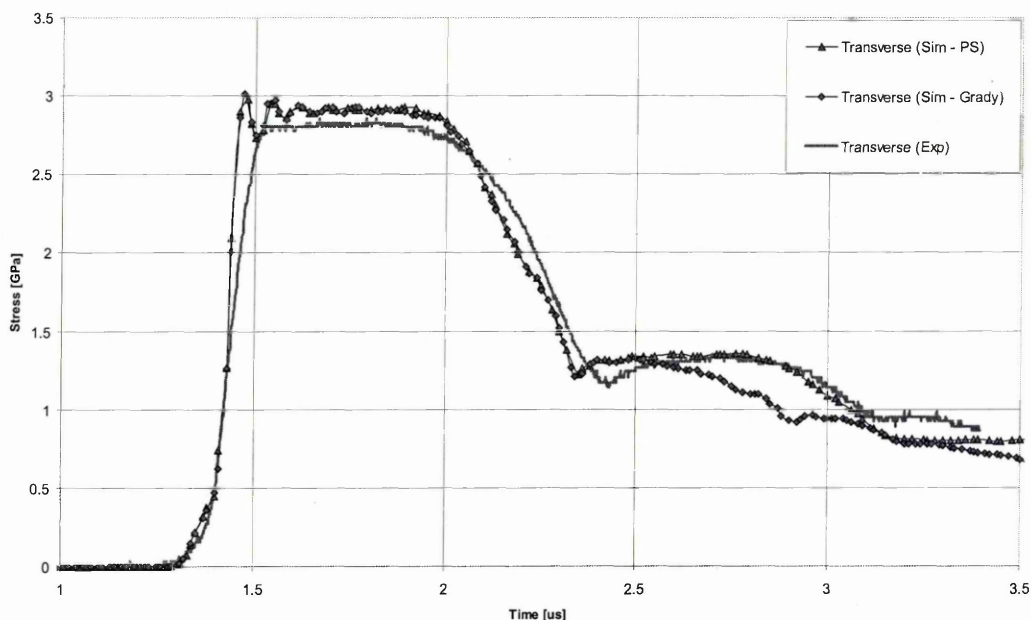


Figure 5.19: 895 m/s Impact – Transverse Direction

In Figure 5.16 and Figure 5.17 the simulated gauge traces at the higher impact velocity of 450 m/s (at two times the HEL) clear spall signals are resolved. Again one can observe a difference in HEL, depending on the direction of impact. The general pulse shape is in good agreement with the experiment. The Hugoniot stress levels are 1.28 GPa for the simulation and 1.3 and 1.4 GPa, depending on direction of impact, in the experiment. Again this is in very good agreement. One can clearly see pull back signals in both traces which indicate spall. The measured values of the pull back signal (spall) in the experiment at  $450 \text{ m s}^{-1}$ , 0.31 GPa in the longitudinal orientation and 0.45 GPa in the short transverse. The values of the pull back signal in the simulation using the principal stress criterion are not really affected by the direction of impact, and are around 0.2 GPa for both directions (ie. errors of 50% to more than 100%). The pull back signal when Grady's spall criterion is used are even smaller, 0.10 and 0.16 GPa, so the errors are larger. The positive point in using Grady's model is the point at which the pullback signal starts is clearly different with direction of impact. This is in line with what is observed in the experimental results. The results for the pressure based spall criterion are very similar to the ones obtained with the maximum principal stress spall criterion and are not shown in order to keep the figures readable. It is clear that the current simple spall models are inadequate to qualitatively model spall behaviour.

At the  $895 \text{ m s}^{-1}$  impacts the picture is similar (Figure 5.18 and Figure 5.19). The general shape of the pulse agrees quite well with the experiment, the Hugoniot stresses compare quite well. In the experiment values of 3.25 GPa in the longitudinal direction and 2.8 GPa in the transverse direction are recorded. The simulation results predict a value of 2.9 GPa (ie. errors of 16% and 3%). For the comparison of the spall signals the situation is similar to the 450 m/s impact. In the simulation the pull back signals are 0.15 GPa using the principal stress and 0.13 GPa using Grady's spall criterion, while the

experiments record 0.37 GPa in the longitudinal orientation and 0.18 GPa in the short transverse. The main point here is that the pull back signals here compare reasonably well with the experimentally recorded pull back signal in the transverse direction. With the simple spall models that are being used one can not expect to observe the complex evolution of spall strength with a change stress levels and strain rates. These trends can not be explained by first order spall theory, and further work is required to understand this process. Again, the pull back signals when Grady's spall criterion is used are smaller than the ones obtained using the maximum principal stress criterion.

Overall, the simple spall models that have been evaluated are capable of predicting whether or not spall occurs. But they are not capable of accurately simulating the pull back signals that result from spall. In order to be able to simulate these signals accurately further work is required in the development of spall models. Especially if one is to simulate the complex spall behaviour observed in Al 7010-T6, where the spall strength increases with strain rate, but reduces with stress levels. This will require further work both on the experimental and constitutive modelling level.

#### 5.5.5 Conclusion

Plate impact simulation have been performed on the aluminium alloy 7010, in the peak-aged condition, where the HEL and spall strength have been compared with the experimental results. It was found that the current model is capable of simulating the higher HEL in the longitudinal direction compared to the short transverse, following the trends of the quasi-static yield stress. The Hugoniot stress levels are predicted to a high level of accuracy, as well as the general shape of the pulse and pulse width. The spall models that were tested are able to qualitatively predict spall. However they are not capable of predicting the pull back stress level at any degree of accuracy. If simulation is to be used to better understand the material behaviour under impact loading further work will be required in order to improve the simulation of spallation of the material. More complex spall models are required, which will require further work both on the experimental and constitutive modelling level.

## Chapter 6 Water Impact of Rigid Cylindrical Specimen

### 6.1 Introduction

If a coupled FE-SPH model approach is to be used to simulate the impact on water of collapsing structures, the coupled FE-SPH code should be capable of dealing with the simpler case of the water impact of rigid shapes on water. Hence as a first step the water impact of a rigid cylinder on water will be used to validate the coupled FE-SPH code. The simulation results will be compared to experimental results.

#### 6.1.1 Test Setup

The drop test data of rigid cylinders onto water that will be used in this chapter (Figure 6.1) where performed by Politecnico di Milano as part of the CAST project [ 1 ]. The acceleration time histories that where recorded during these series of tests will be compared to validate the simulation results.

The cylindrical specimen were made of plies of evaporated beech. The dimensions of the specimen (see Figure 6.2) are a radius of 121mm, and a length of 1000mm. The surface of these cylindrical blocks was covered in glycol reinforced with fibre glass fabric. The total weight of the test specimen was 20 kg. Accelerometers where mounted on the top of the specimen at the two ends of the cylinder (see drawing in Figure 6.2). The sampling frequency was 12500Hz and CFC1000 filtering is applied to the time-history data.

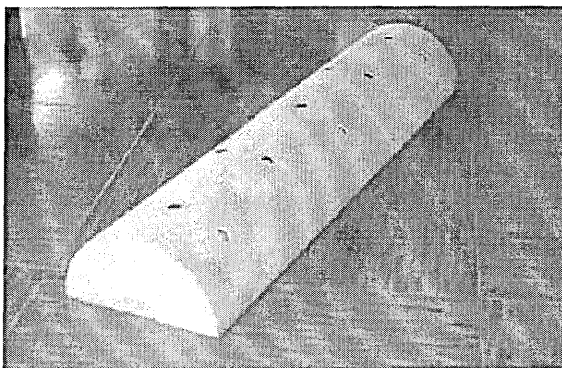


Figure 6.1: Rigid Cylinder used in Water Impact Test, from [ 1 ]

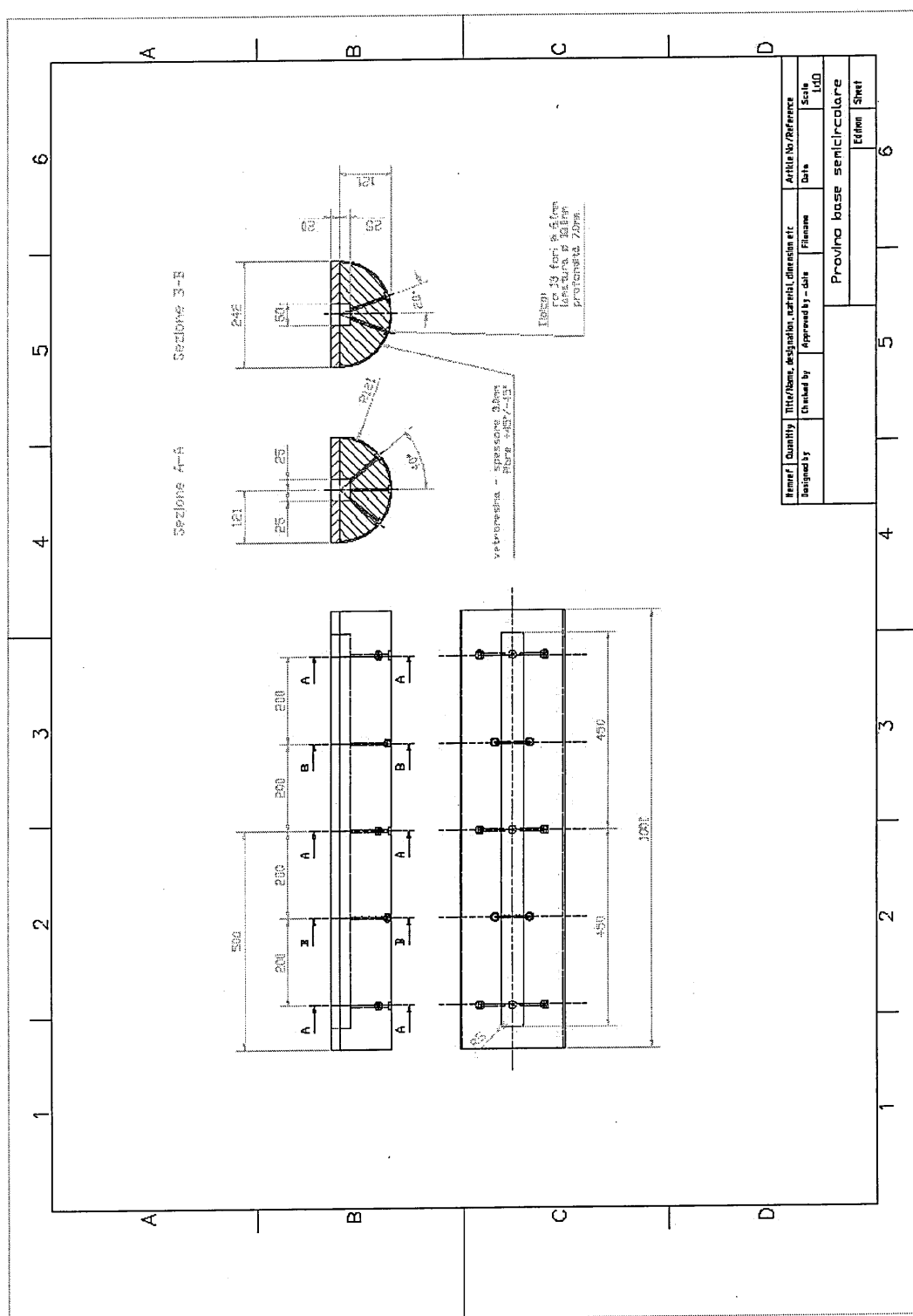


Figure 6.2: Technical Drawing of Rigid Cylinder used in Water Impact Test, from [ 1 ].

The impacts speeds where varied by changing the drop height, and tests where performed at three different velocities, 10.6 m/s, 7.8 m/s and 3.3 m/s. Figure 6.3 shows

two high speed camera pictures of the experiment. The resulting measured accelerations are summarised in Table 6.1.

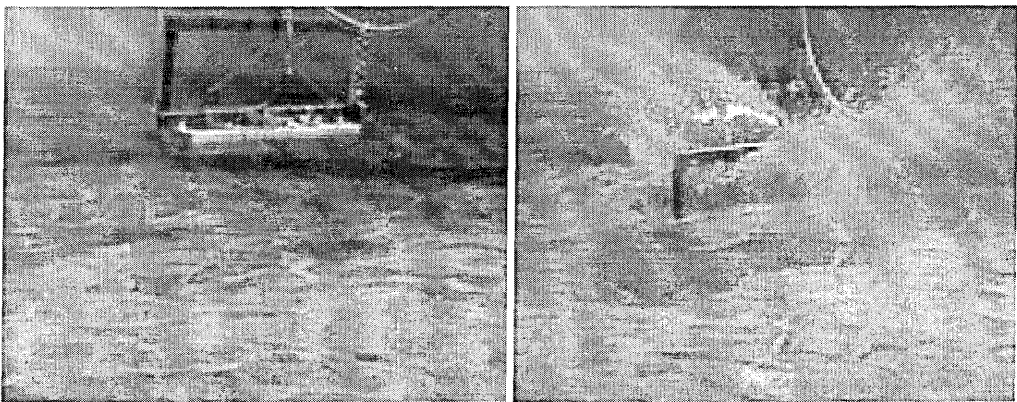


Figure 6.3: Water Impact of Rigid Cylindrical Specimen, Experiment, [ 1 ])

Velocity [m/s]	Acceleration #1 [m/s]	Acceleration #2 [m/s]
10.6	81.0	82.9
7.8	41.5	41.1
3.3	20.2	22.8

Table 6.1: Measured Acceleration during Water Impact Experiments.

6.2 Simulation Results

6.2.1 Model Description and Sensitivity Study

A model of the cylinder impact on water was built that contains 3740 solid (hexahedral) elements and 12800 SPH particles (see Figure 6.4 and Figure 6.6). The rigid cylinder is modelled with hexahedral elements, while the water is modelled with SPH particles. Because of the symmetry of the problem only a quarter of the cylinder is modelled, and symmetry planes are defined to complete the model geometry. A volume of water of 60x25x30cm was discretised, where 30cm is the depth of water block. The particle spacing was 1.5cm. An elastic material model was used for the cylinder (Young’s modulus of 6.0 MPa), a fluid material model (no shear stresses) with equation of state was used to model the water. The interaction between the cylinder and the water is accounted for by using the coupling algorithm described in Chapter 4, and the contact parameters where  $K = 2.0$  and  $n = 4.0$ . In order to determine the optimum model setup several models where created to check the influence of different modelling assumptions:

- ▶ Two equations of state (EOS) where tested (Gruneisen and Murnaghan).
- ▶ The contact stiffness was varied.
- ▶ The mesh density was increased.
- ▶ The volume of water was increased.
- ▶ Realistic material properties of wood where used.
- ▶ The influence of the ratio between particle spacing and smoothing length was checked.

These parameters were varied for the simulation of the 10.6m/s impact. The first model uses a Grüneisen equation of state, and the Young's modulus of the cylinder is defined lower than the actual value for beech, such that it behaves approximately as a rigid body but without affecting the timestep. A Young's modulus of 0.6 GPa was used. The values for the Grüneisen equation of state for the water model are shown in Table 6.2.

Parameter	Value
$C_0$	0.14829
$S_1$	2.03670
$S_2$	0.00000
$S_3$	0.00000
$\Gamma$	1.00000
$a$	0.50000

Table 6.2: Grüneisen EOS parameters for water

The results obtained with this initial model are shown in Figure 6.4 to Figure 6.7, these figures show a half-model for clarity, but this was achieved by mirroring the model in the post-processor. The acceleration time histories of this simulation are shown in Figure 6.9. The time histories were recorded at three different points. The location of these points is shown in Figure 6.8. The point labelled 576 is positioned where one of the accelerometers was fixed. The point labelled 181 is located at the centre of mass, while point 544 is near the centre of mass of the cylinder, but away from the symmetry planes. This last node allows to check if there is any influence from the use of symmetry planes or possible bending of the cylinder on the results. The peak acceleration is around 102.5g for all three points. The pulse length of around 6ms is shorter than the 10ms pulse which is observed in the experiment.

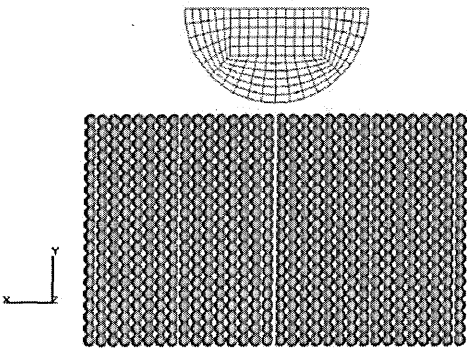


Figure 6.4: Water Impact of Rigid Cylindrical Specimen, Side view of Model (t = 0ms).



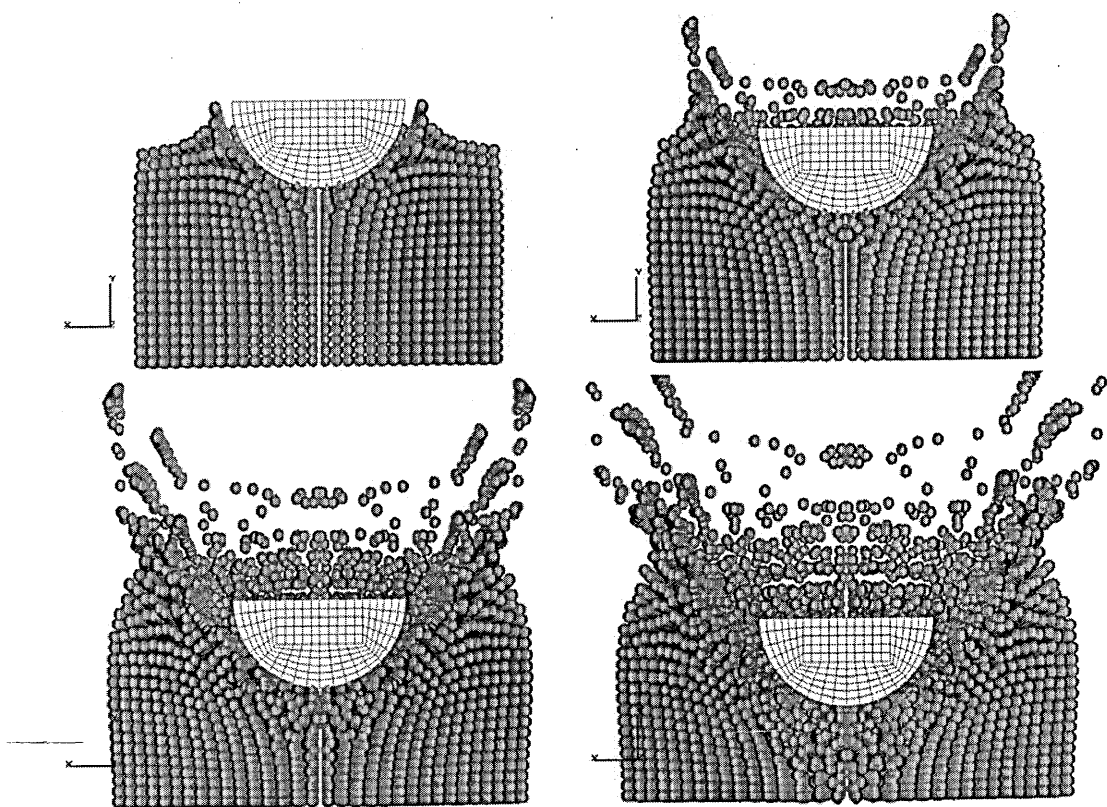


Figure 6.5: Water Impact of Rigid Cylindrical Specimen, Side view of Model ( $t = 10\text{ms}$ ,  $20\text{ms}$ ,  $30\text{ms}$ ,  $40\text{ms}$ ).

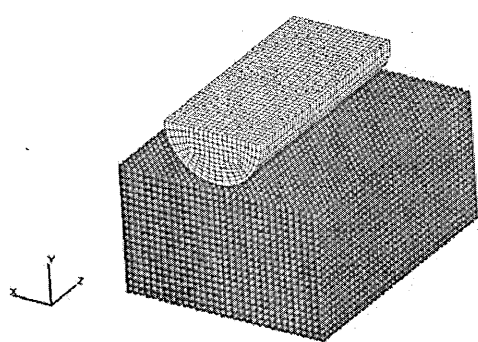


Figure 6.6: Water Impact of Rigid Cylindrical Specimen, Isometric of Model ( $t = 0\text{ms}$ ).

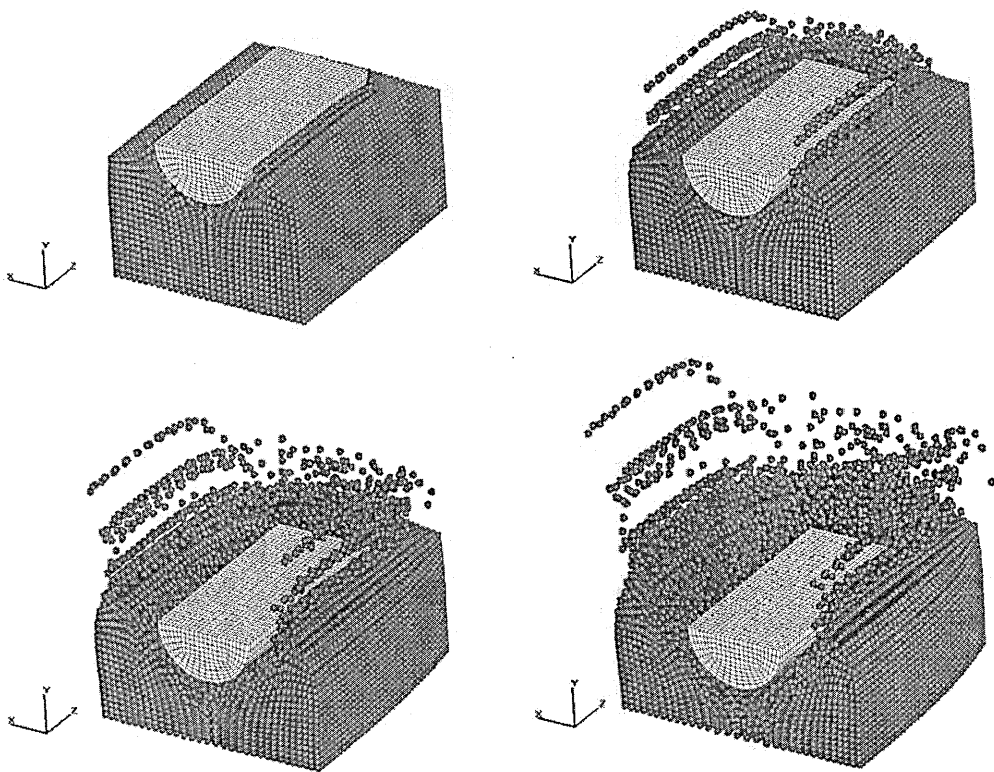


Figure 6.7: Water Impact of Rigid Cylindrical Specimen, Isometric of Model ( $t = 10\text{ms}$ ,  $20\text{ms}$ ,  $30\text{ms}$ ,  $40\text{ms}$ ).

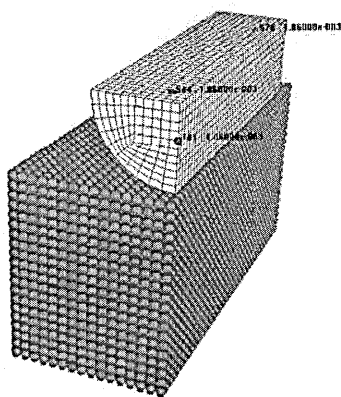


Figure 6.8: Water Impact of Rigid Cylindrical Specimen, location of time history nodes.

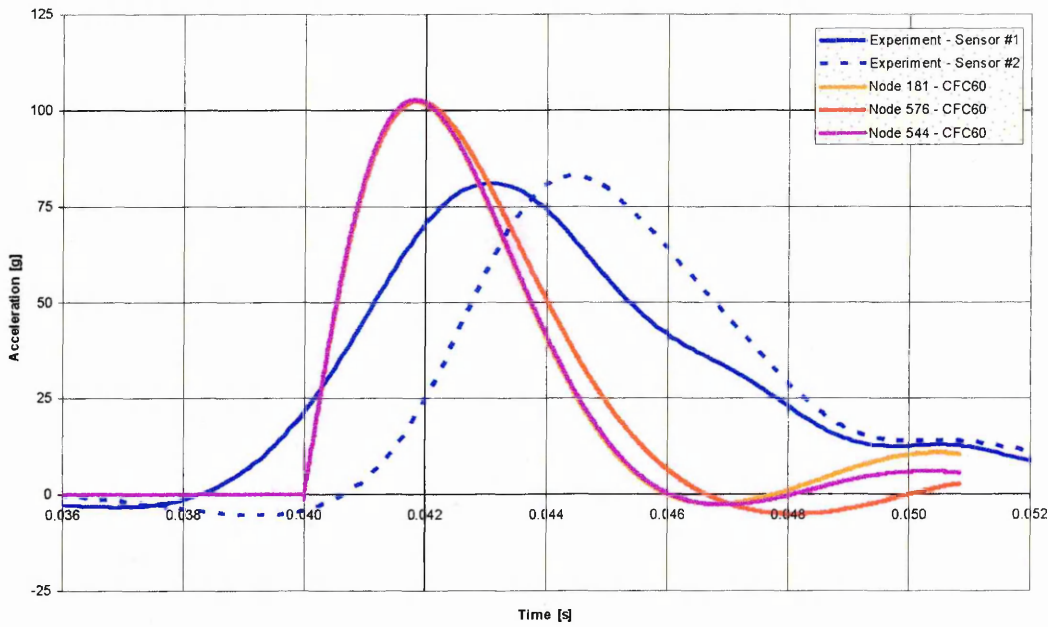


Figure 6.9: Initial model results, acceleration time histories.

### 6.2.2 Influence of EOS

In order to check the influence of the EOS for water on these results the simulation was repeated with a second EOS, the Murnaghan EOS. The Murnaghan equation of state ensures that the fluid has an artificially increased compressibility such that it behaves quasi-incompressible, and relates the pressure to the density in the following manner:

$$p = B \left[ \left( \frac{\rho}{\rho_0} \right)^\gamma - 1 \right], \quad (\text{Eq. 6.1})$$

where  $B$  and  $\gamma$  are user defined constants. This EOS can be used in situations where the flow velocity is well below the physical sound speed, and compressibility effect are of minor importance. The speed of sound is determined by the choice of the parameter  $B$ . Generally one chooses  $B$  such that:

$$B > \frac{100\rho v^2}{\gamma} \quad (\text{Eq. 6.2})$$

This ensures the required incompressibility, but results in a sufficiently large timestep. The speed of sound calculated from this EOS is not related to the actual speed of sound of the material, hence the timestep will be much larger than the timestep calculated with the actual speed of sound of the fluid. Therefore one can simulate problems involving incompressible fluids using this EOS with a larger timestep than if the timestep was based on the speed of sound. This is acceptable as long as the problem is not dominated by wave propagation effects.

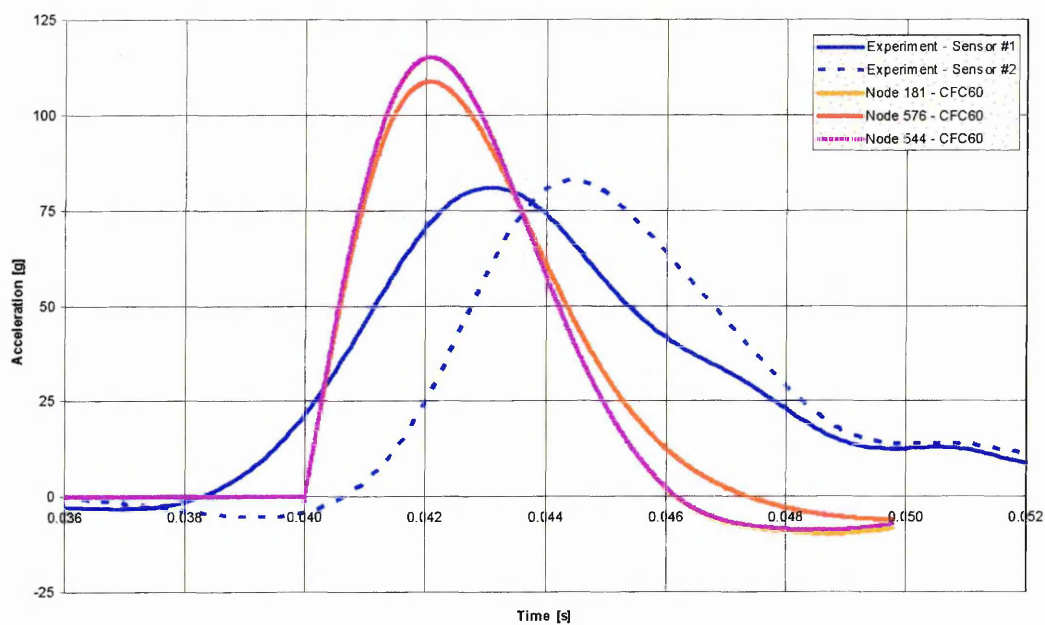


Figure 6.10: Murnaghan EOS results ( $B = 0.000016$ )

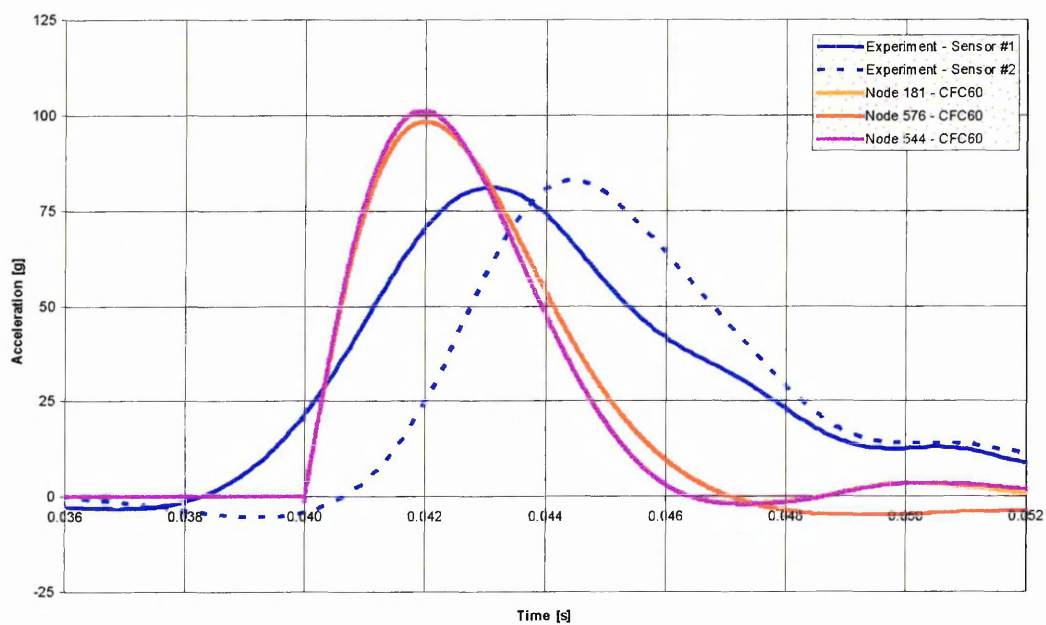


Figure 6.11: Murnaghan EOS results ( $B = 0.000060$ )

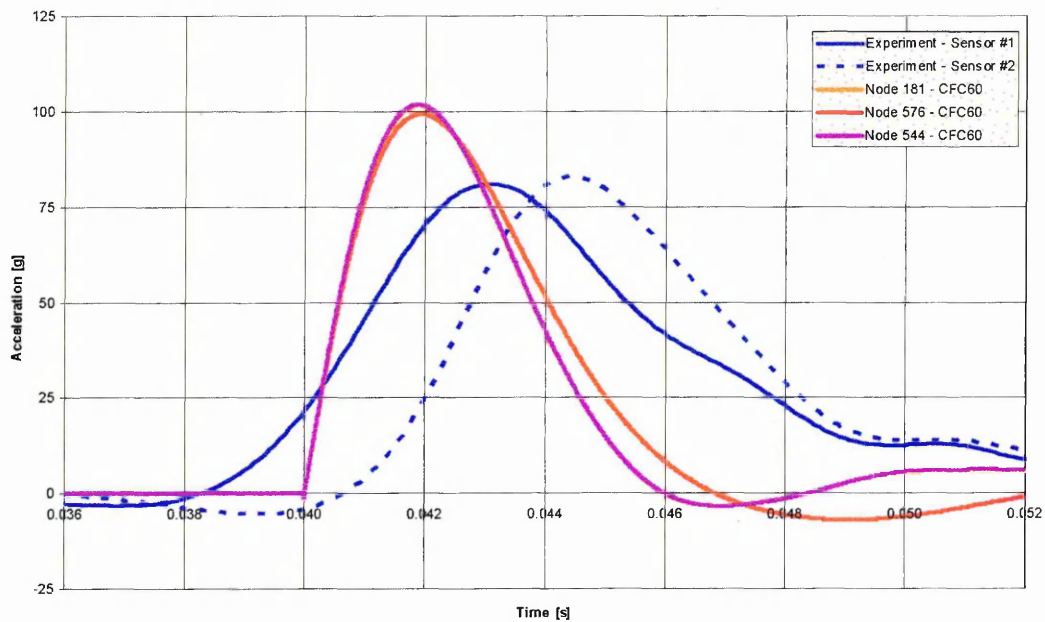


Figure 6.12: Murnaghan EOS results ( $B = 0.000180$ )

A major problem with the use of equations of state is the availability of material data. Most of the publicly available data is on the behaviour of solids. The Gruneisen EOS is widely used in hypervelocity impact calculations, and water material data is available. When using the Murnaghan EOS no problems of availability of material data arise because the input parameters for this EOS are dependent only on the impact velocity, and the assumption of incompressibility.

The results for the model with different EOS are shown in Figure 6.10, Figure 6.11 and Figure 6.12. The results in Figure 6.10 use values of 0.000016 for  $B$  and 7.0 for  $\gamma$ . The resulting acceleration time history is higher than the reference simulation which uses the Gruneisen equation of state. The acceleration peaks are at 115.2g for nodes 181 and 544, and at 108.8g for node 576. Increasing the value of  $B$  reduces the compressibility. With a higher value of  $B$  of 0.00006 the results are similar to the reference configuration using a Gruneisen equation of state. The acceleration peaks are at 101.2g for nodes 181 and 544, and at 98.3g for node 576. A further increase of  $B$  to 0.000180 does not show any significant change in the results (Figure 6.12).

### 6.2.3 Influence of Contact Stiffness

A second parameter which was tested was the influence of the contact stiffness. The results for the model with different contact stiffness are shown in Figure 6.13 and Figure 6.14, and show an increased contact stiffness and reduced contact stiffness respectively. Figure 6.13 shows that for the increased contact stiffness ( $K = 20.0$ , as opposed to 2.0 in the reference configuration) the acceleration peaks are much lower at 84.4g for nodes 181 and 544, and at 85.3g for node 576. Figure 6.14 shows the results for a reduced contact stiffness of  $K = 0.2$ , the acceleration peaks are much higher at

128.3g for nodes 181 and 544, and at 126.1g for node 576. However on the plots of the reference configuration (with contact stiffness parameter  $K = 2.0$ ) and the simulation with the increased contact stiffness of  $K = 20.0$  an unphysical separation of the water from the cylinder after contact could be observed. This indicates that the contact stiffness is too high, and also explains why the acceleration peaks are lower with increasing contact stiffness. This would also explain why the accelerations do not show the constant acceleration of around 10g observed after the initial acceleration peak in the experimental results. If the particles separate from the surface no contact forces will be present and the acceleration will be around zero. When the reduced contact stiffness of  $K = 0.2$  is used this unphysical separation is not present, but the particles start to penetrate the cylinder surface. In order to overcome this problem an option was added to the contact algorithm to explicitly enforce the persistency of the contact and is expressed as (see Chapter 3):

$$\mathbf{t}_n \cdot \dot{\mathbf{g}} = 0$$

The implementation of this condition is straightforward as it simply requires the contact force to be zero if the penetration is decreasing. The disadvantage of this option is that it introduces energy dissipation.

The reason that it is very difficult to find an appropriate contact stiffness without enforcing the persistency condition may be explained by considering two instants during the impact, immediately after the impact and during the steady state ‘drag’ phase. Upon impact the contact forces will be very high because of the velocity difference between the cylinder and water surface. At a later stage the cylinder will reach a steady state where it pushes the water in front of it. During that stage the contact forces should be smaller. To improve the performance of the contact algorithm in dealing with these two cases and without enforcing the persistency condition one could add a pressure dependent term in the contact potential. The pressure in the materials will be high during the initial impact, but the pressure in the materials will be low when the cylinder is moving through the water. A pressure dependent term in the contact force would result, for the same penetration, in higher contact forces when the pressure is high immediately after contact.



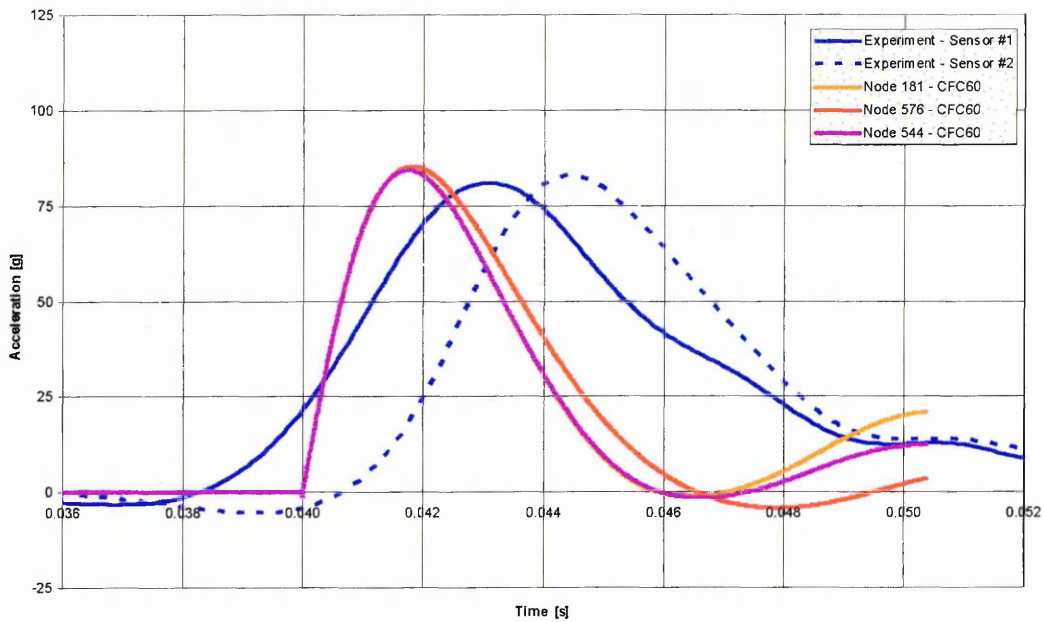


Figure 6.13: Increased Contact Stiffness Results

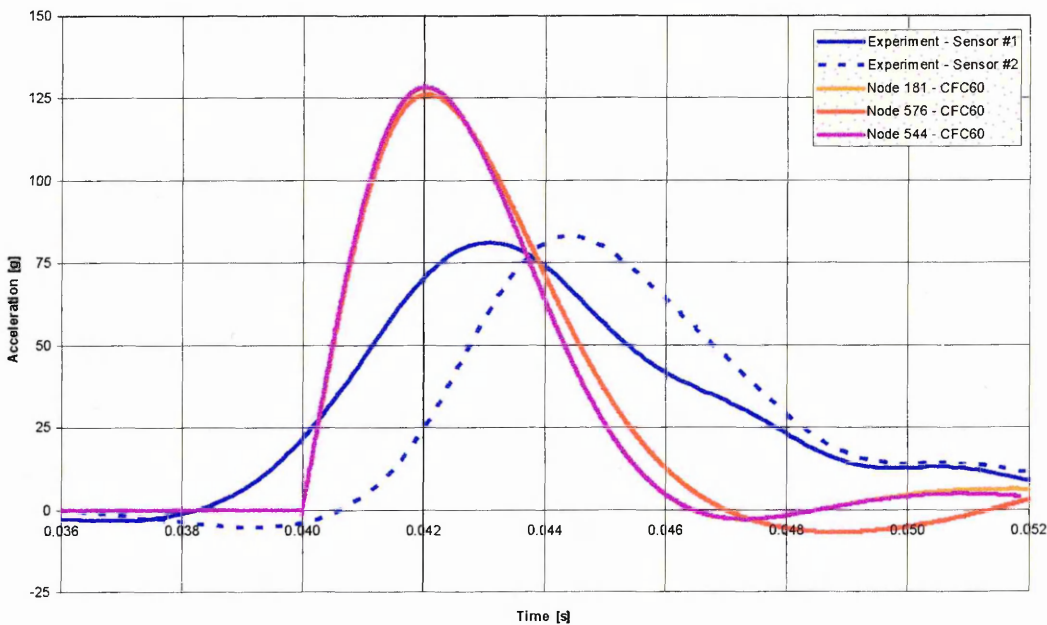


Figure 6.14: Reduced Contact Stiffness Results

#### 6.2.4 Influence of the discretisation density

Another parameter which may influence the results is the discretisation density. In order to check the influence of this parameter on the results a finer grid was constructed. This finer grid has 45000 particles in the volume of water. The use of 45000 particles corresponds to a particle spacing of 1.0 cm, ie. two thirds the spacing of the original

discretisation. The results of this simulation are shown in Figure 6.15. The acceleration peaks are slightly lower at 99.9g for nodes 181 and 544, and at 98.9g for node 576.

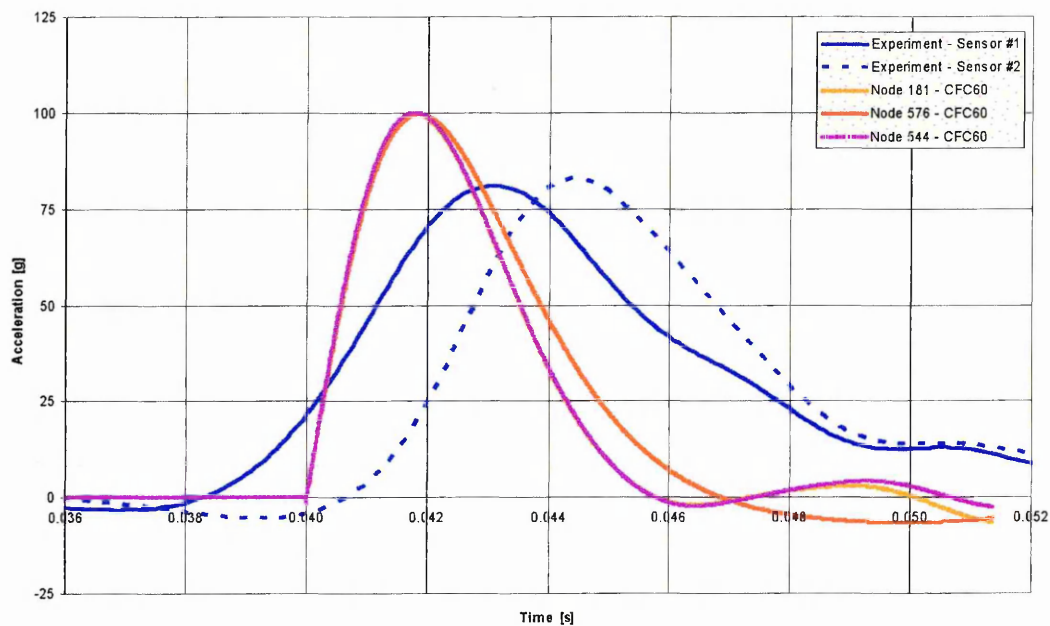


Figure 6.15: Finer Mesh Results

### 6.2.5 Influence of the volume of water discretised

Another parameter which may influence the results is the volume of water that is modelled. In order to check the influence of this parameter on the results an increased volume of 75x40x50 cm was discretised. This finer grid has 42900 particles in the volume of water. The results of this simulation are shown in Figure 6.16. The acceleration peaks are slightly lower at 97.8g.

### 6.2.6 Influence of Smoothing Length

In the initial simulation a smoothing length of 1.7cm is used, the particle spacing is 1.5cm. A second simulation was performed where the smoothing length was equal to the particle spacing. The resulting acceleration time histories are shown in Figure 6.17. The acceleration peaks are much higher at 119.8g for nodes 181 and 544, and at 119.3g for node 576. This shows that it is necessary to use a smoothing length which is larger than the particle spacing to obtain more realistic results.

### 6.2.7 Influence of Young’s Modulus of Wood

The final variable that was tested for its influence on the simulation results was the elastic modulus of the cylinder. The reference simulation uses a Young’s modulus of 6.0 MPa, whereas the Young’s modulus of beech wood is around 12.0 GPa [ 88 ]. If this more realistic Young’s modulus is used the resulting acceleration peak is slightly



lower at 101.5g (as opposed to 102.5g). This indicates that the precise value of the Young's modulus for wood is a minor factor in this type of simulation.

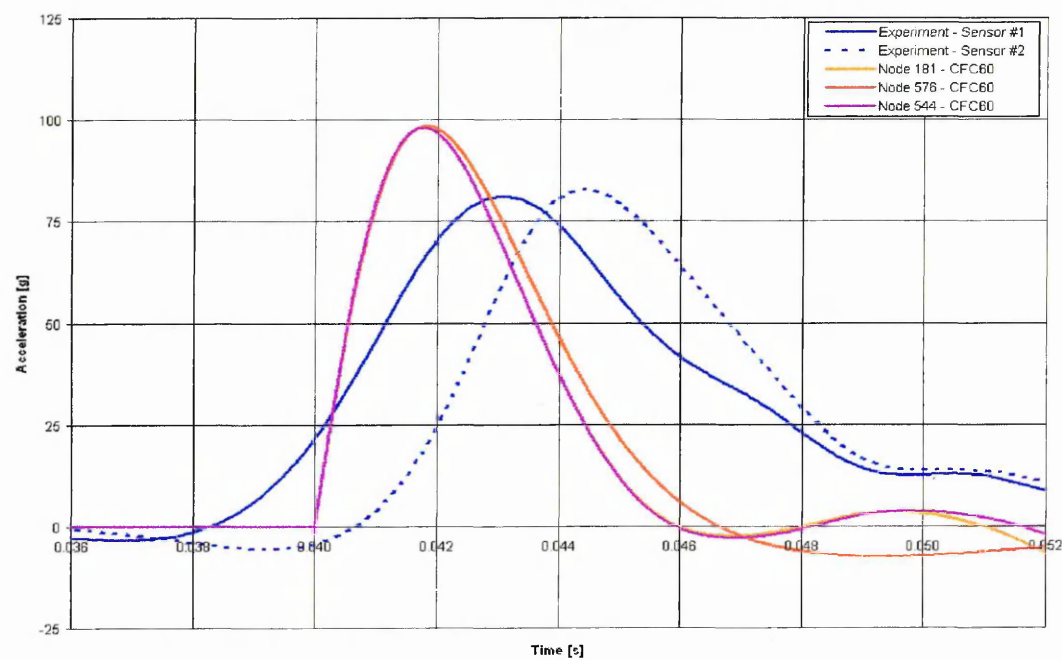


Figure 6.16: Increased Volume of Water Results

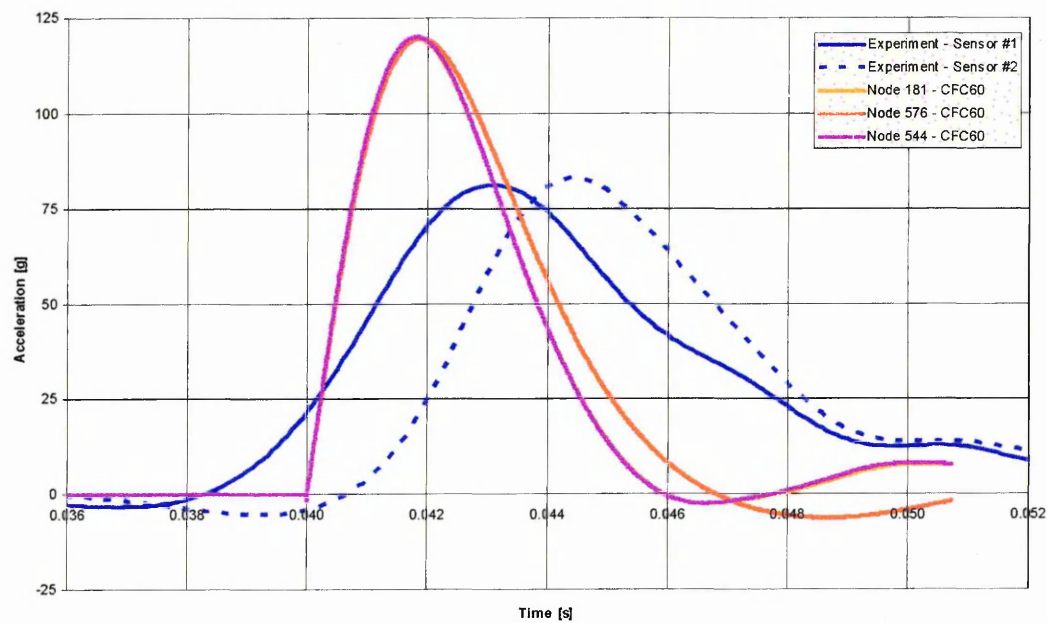


Figure 6.17: Smoothing length equal to particle spacing results

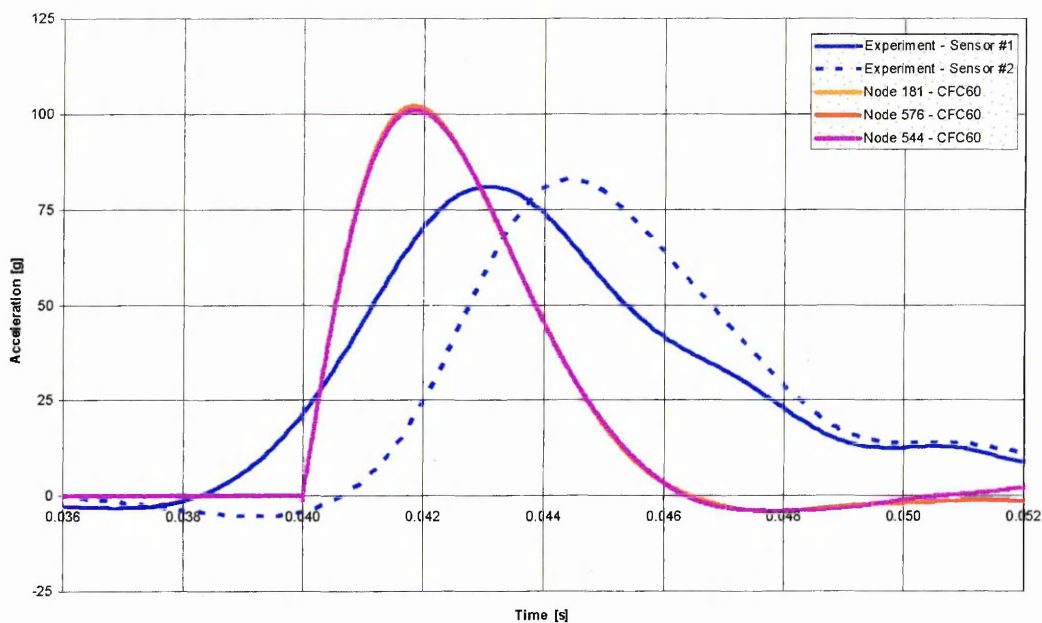


Figure 6.18: Properties of wood

### 6.2.8 Comparison of Experimental and Simulation Results

From the sensitivity study it can be concluded that:

- ▶ The use of a Murnaghan EOS does not offer any benefit to the modelling of water as no improvement in results could be obtained.
- ▶ The value of the contact stiffness has a big influence on the acceleration levels.
- ▶ The use of a finer mesh results only in a small reduction of the acceleration peak.
- ▶ The volume of water of 60x25x30 cm discretised is adequate.
- ▶ It is important to use a value for the smoothing length which is bigger than the particle spacing.
- ▶ The influence of the material properties of the cylinder is minimal.

Based on these results it was decided to simulate the cylinder impacts on water at the three velocities of 10.6, 7.8 and 3.3 m/s with one modification to the initial model. This modification consisted of enforcing the 'contact persistency' condition to avoid artificial separation or interpenetration. The simulations time was increased to 40ms to allow for the comparison of the g-levels after the initial peak.

Figure 6.19 shows the comparison with a finite element model results for the fastest impact of 10.6 m/s. The only difference between the coupled FE-SPH model and the FE model is that every particle was replaced with one element. The magnitude of the initial peak is in much better agreement using the coupled DYNA3D-SPH approach., and is between 83.5 and 84.0 g (2.4% error). In the case where the water is modelled with finite elements the magnitude of the initial peak is 70.9 g (13.5% error). However the pulse length is 11 ms when using finite elements to model the water, as opposed to

7 ms, which is in better agreement with the experimental data where the pulse length is 9.7 ms. Figure 6.20 shows the velocity time history of the cylinder for both simulations. After 10ms the velocities are very similar. This means that the energy which has been transferred into the water is the same.

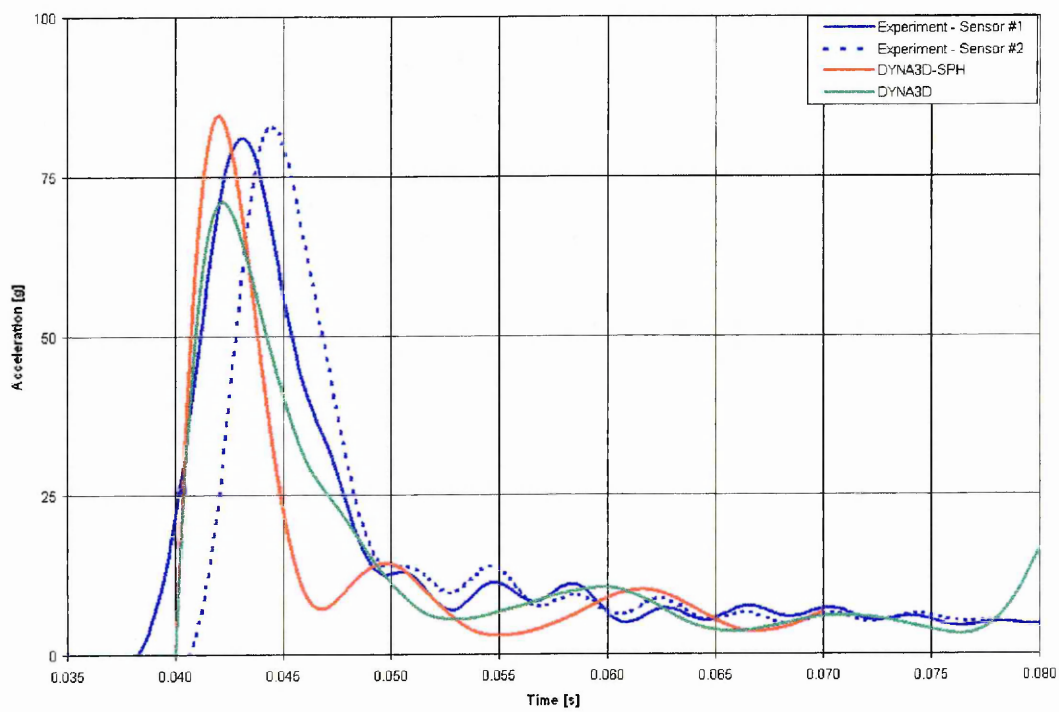


Figure 6.19: Comparison of Experimental and FE and SPH Simulation Results of Cylinder Impact at 10.6m/s

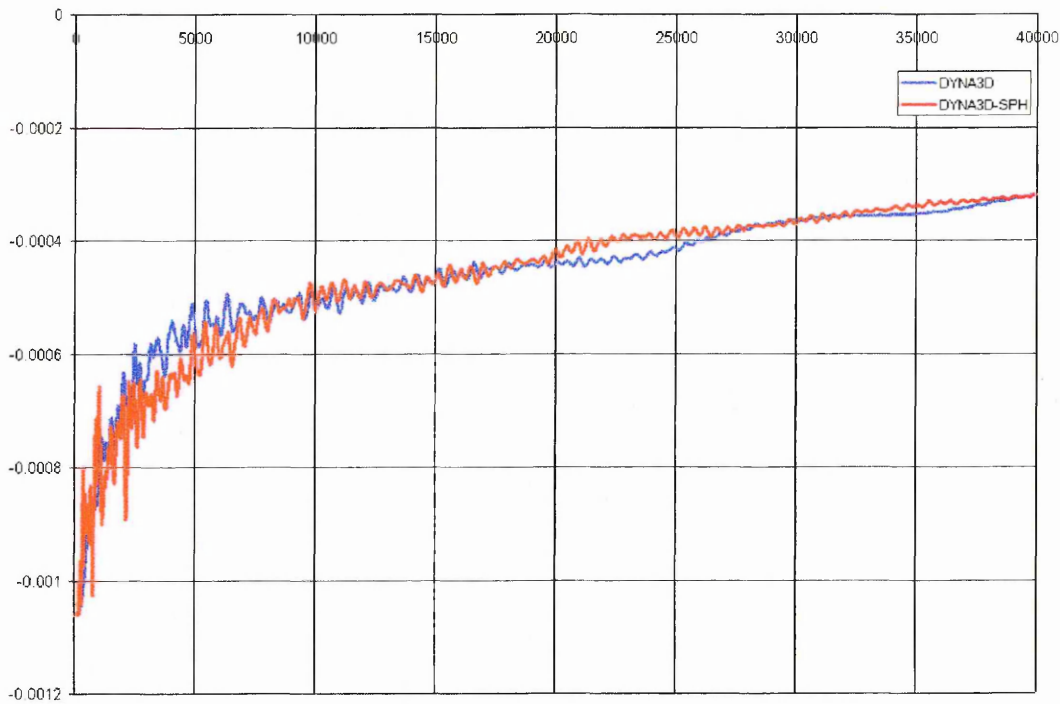


Figure 6.20: Comparison of FE and SPH Velocity Time Histories of Cylinder Impact at 10.6m/s

The results of the water impact simulations at the three impact speeds of 3.3, 7.8 and 10.6 m/s are shown in Figure 6.21 to Figure 6.23, and the acceleration levels are summarised in Table 6.3. Figure 6.21 shows the results for the slowest impact of 3.3 m/s. The simulated acceleration levels 19.1 g to 20.5 g (depending on the position) are in excellent agreement with the experimentally recorded accelerations (21g). The shape and pulse length is in good agreement also. Due to the oscillations in the simulated time histories it is difficult to comment on the steady decrease of acceleration from 3.4g to 2.1 g observed in the experiment. In the simulation the magnitudes appear to be lower, decreasing from 2.2 g to 1.1 g. This may be due to the fact that gravity has been ignored. The divergence of the simulation time histories at the end of the time histories is due to the combination of high oscillation in the data and the data initialisation algorithm used in the CFC filters.

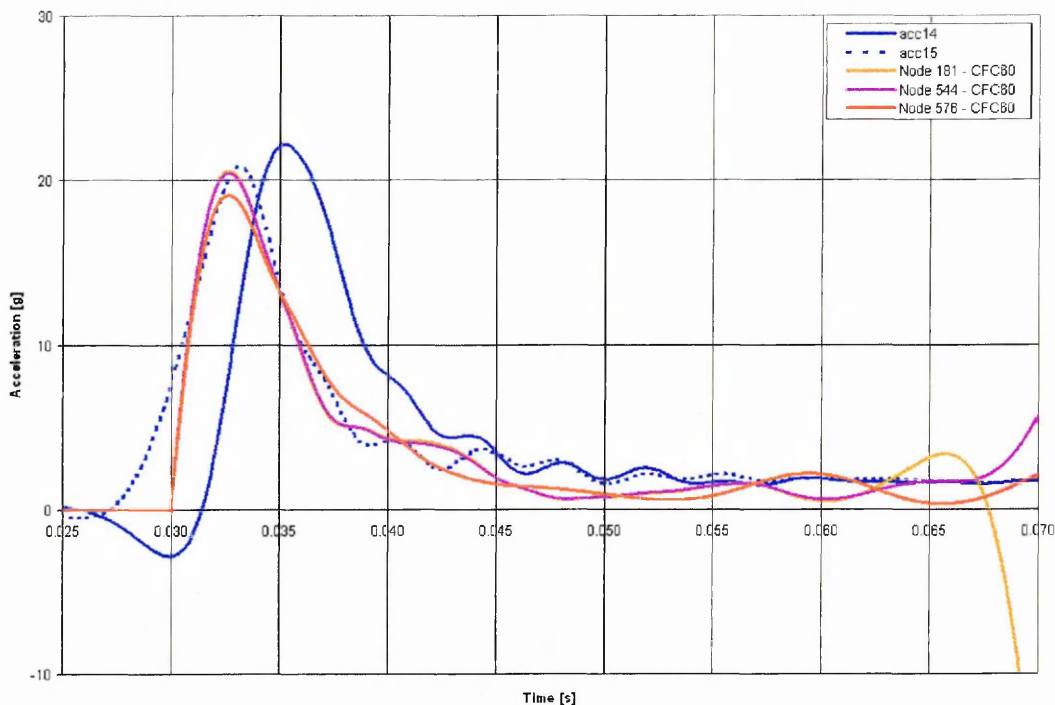


Figure 6.21: Comparison of Experimental and Simulation Results of Cylinder Impact at 3.3m/s

Figure 6.22 shows the results for the impact of 7.8 m/s. The simulated acceleration levels are far higher (61.5 to 62.3g) in the simulation than the experimentally recorded accelerations (41g). The acceleration pulse also is shorter in duration than in the experiment. The explanation for this discrepancy may be explained by noting that in the experiment the cylinder impacted the water surface at a 2.7° inclination. The effect of this would be that one end of the cylinder starts decelerating before the other end, which is clearly the case in this experiment (see Figure 6.22). Due to rigid body rotation the opposite end of the cylinder could in fact accelerate, which is what is observed in Figure 6.22. This gradual impact due to the inclination could result in lower acceleration peaks and explain the big difference between experimental and simulation results.

Figure 6.23 shows the results for the fastest impact of 10.6 m/s. The simulated acceleration levels of 83.5 to 84.0 g are in good agreement with the experimentally recorded accelerations (81.0 and 82.9 g). The pulse length of 7ms is shorter than the experimentally recorded pulse length of 9.7 ms. The steady decrease of acceleration from 10g to 4.5 g observed in the experiment is to some extent also present in the simulated time histories. However there are large oscillations in the signal and the magnitudes are lower, decreasing from 8.2 g to 5.3 g.

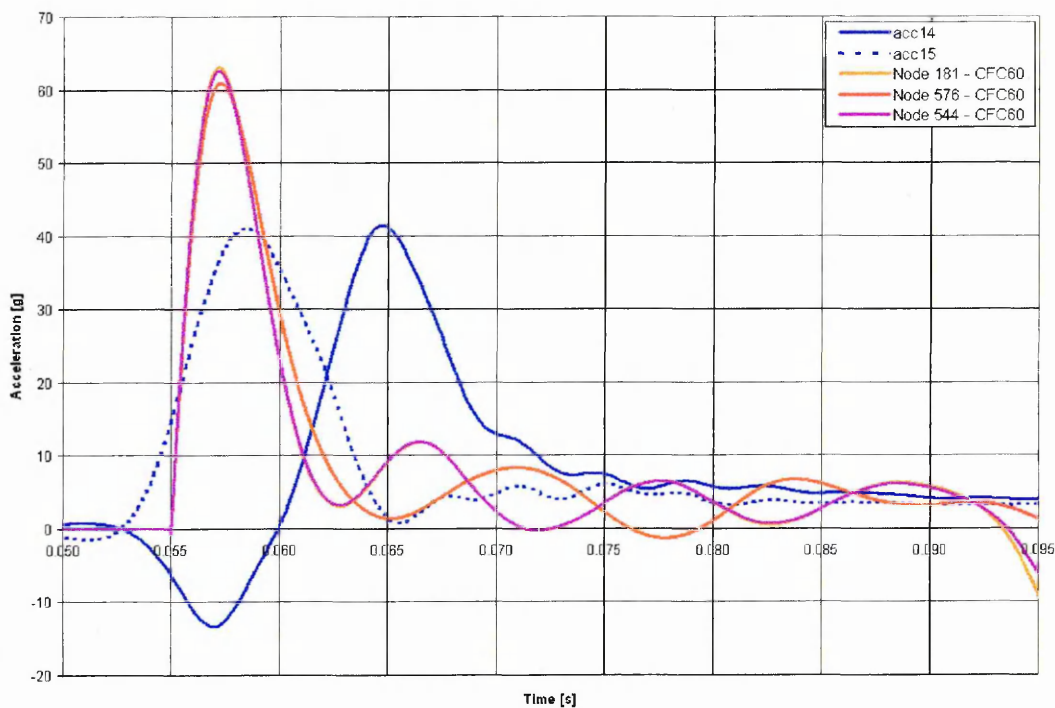


Figure 6.22: Comparison of Experimental and Simulation Results of Cylinder Impact at 7.8m/s

From the good agreement with experiments of the 3.3 m/s and 10.6 m/s simulation results it can be concluded that a coupled FE-SPH modelling approach of water impact offers a promising alternative to complete FE models. The discrepancies in the intermediate velocity impact of 7.8 m/s may be explained by the 2.7° inclination of the cylinder in the experiment. Other factors that may influence the results are gravity, the presence of an air cushion on impact, and cavitation on the cylinder surface.



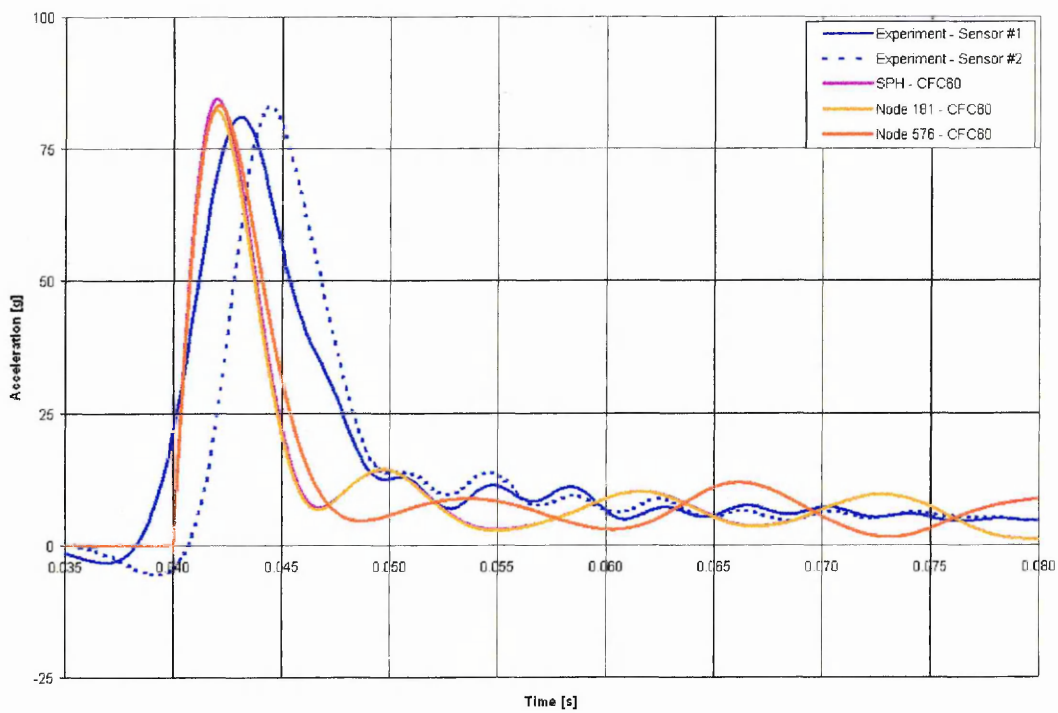


Figure 6.23: Comparison of Experimental and Simulation Results of Cylinder Impact at 10.6m/s

Velocity [m/s]	Acceleration #1 [g]	Acceleration #2 [g]	Simulation [g]
10.6	81.0	82.9	83.5 - 84.0
7.8	41.5	41.1	61.5 - 62.3
3.3	20.2	22.8	19.1 - 20.5

Table 6.3: Comparison of Simulated Acceleration Peaks to Experimental Data

## **Chapter 7 Applications – Water Impact of Metallic Structures**

### **7.1 Subfloor Description**

The driving force behind the improvements to LLNL-DYNA3D that were presented in the previous chapters was the simulation of fluid-structure interaction between fluids and deformable structures, in particular the collapse of aircraft on water. The goal of this chapter is to combine the developments from the previous chapters in the simulation of water impact of a structure representative of a helicopter subfloor. A full validation of the simulation results is not possible due to a lack of experimental data. However the simulation results will demonstrate that a water impact simulation using a coupled FE-SPH approach is capable of dealing with water ingress into the structure. Furthermore, the effect of various parameters, such as metal anisotropy and skin thickness, on the results will be illustrated.

A typical helicopter subfloor is shown in Figure 7.1. The longerons, running along the axis of the helicopter, consist of metal sheets, jointed together with C- or L-section stiffeners to the upper and lower panels. The frames are oriented in the transverse direction to the axis of the helicopter. They consist of holed sheets and L-section stiffeners join the frames to the upper and lower panels. The upper panels form the passenger bay floor and therefore usually consist of a mixed aluminium honeycomb/composite panel or an aluminium/honeycomb/aluminium sandwich panel. The lower panel is part of the outer skin of the helicopter and is made of a metallic sheet. The lateral frames are machined frames as they are connected to the main lifting frame, and hence have to support the engine gearbox assembly. The lateral panels consist of metallic sheets which form the lateral curved skin of the sub-floor box. The components are connected by rivets. The material used in this assembly is aluminium, for example the 2024-T3 alloy.

### **7.2 Modelling Approach**

In order to test the FE-SPH coupling in combination with the anisotropic plasticity model a model was constructed of a structure representative of the subfloor described in the previous paragraph. The construction consists of two frames and two outside and one central longeron. These components are connected to lower and top skins. The lateral frames and curved outboard skins are not modelled. The model consists of 9494 shell elements. 125kg of ballast has been added to the four corners (spread over 7x7 elements) to represent the loading from the lift frames and payload. The nodes associated with these elements were also constrained in X, Y and Z directions to move together. This would represent the case of the model being fixed to a rigid trolley in an experiment. The dimensions of the components are listed in Table 7.1. The plate thicknesses of all components are summarised in Table 7.2.



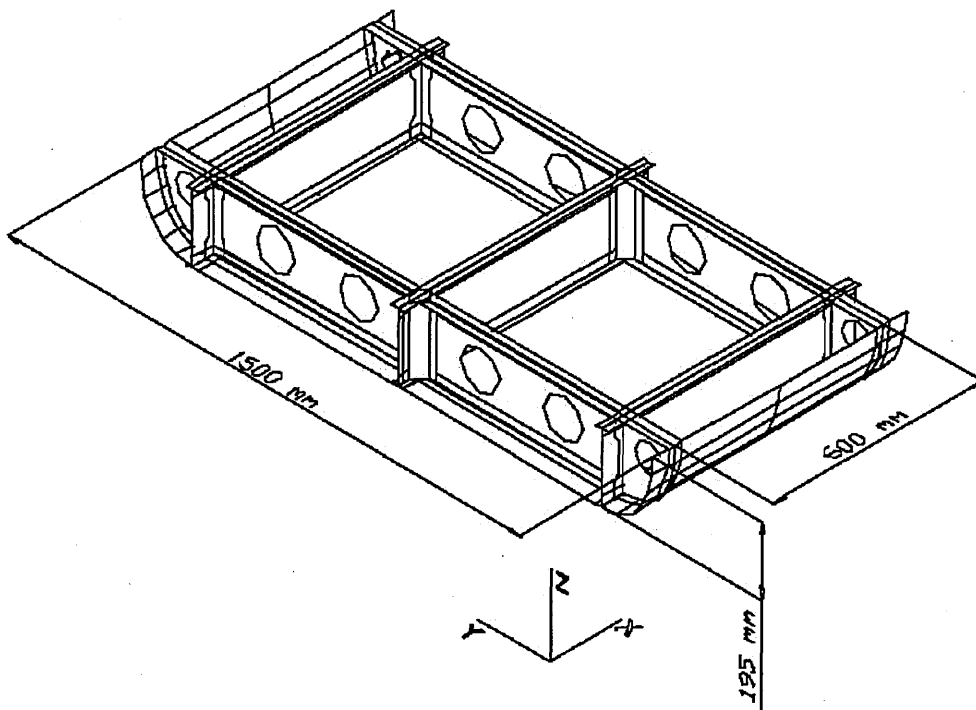


Figure 7.1: Typical Helicopter Subfloor Box.

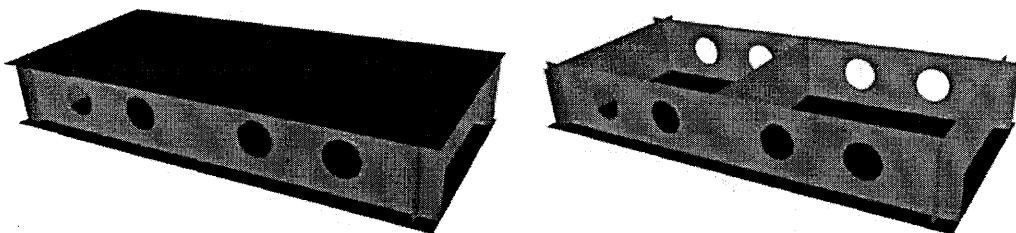


Figure 7.2: Subfloor FE Model

Component	Dimension [mm x mm]
upper skin	1300 x 600
lower skin	1300 x 600
frames	1200 x 200
outside longerons	500 x 200
central longeron	500 x 200
corner stiffeners	50 x 200

Table 7.1: Dimension of Subfloor Model Components

A total of five simulations will be presented in the following paragraphs. Two simulations, one with isotropic material and one with anisotropic material model, of an impact on a rigid surface. These simulations are then repeated for the case of impact on water. Finally a water impact simulation is performed where the lower skin has been

weakened such that it will fail and allow the water to enter the structure. A summary is shown in Table 7.3. In all case the impact velocity is 8m/s. The model does not contain any features to represent rivets, the nodes at intersecting components are simply equivalenced.

Component	Thickness [mm]
upper skin	0.64
lower skin	0.64
frames	1.27
outside longerons	1.27
central longeron	1.27
corner stiffeners	3.50

Table 7.2: Plate Thickness of Subfloor Model Components

Simulation	Material Model	Impact Surface	Lower Skin Thickness [mm]	Failure Strain [%]
1	isotropic plasticity	rigid	0.64mm	-
2	anisotropic plasticity	rigid	0.64mm	-
3	isotropic plasticity	water	0.64mm	-
4	anisotropic plasticity	water	0.64mm	-
5a	anisotropic plasticity	water	0.24mm	-
5b	anisotropic plasticity	water	0.24mm	1% - 15%

Table 7.3: Summary of Subfloor Simulations Setup.

### 7.3 Simulation Results – Rigid Surface

#### 7.3.1 Isotropic Material Model

The first simulation on rigid surface uses an elastic-plastic material model typically used in industry (Material Model 3 in DYNA3D). The model uses Von Mises plasticity theory with combined linear isotropic-kinematic hardening. The material parameters used in this simulation where representative of an aluminium 2024 alloy, see Table 7.4.

Material Constant	Value
Young’s Modulus [GPa]	72.0
Poisson’s Ratio	0.33
Yield Stress [MPa]	600.0
Hardening Modulus [MPa]	667.0

Table 7.4: Material Constants for Al 2024, Simulations 1 & 3.

State plots and time histories at six points on the top skin were recorded. The locations where the time histories are recorded are shown in Figure 7.3. These locations correspond to the four points where the ballast is fixed, and two locations above the central longeron.

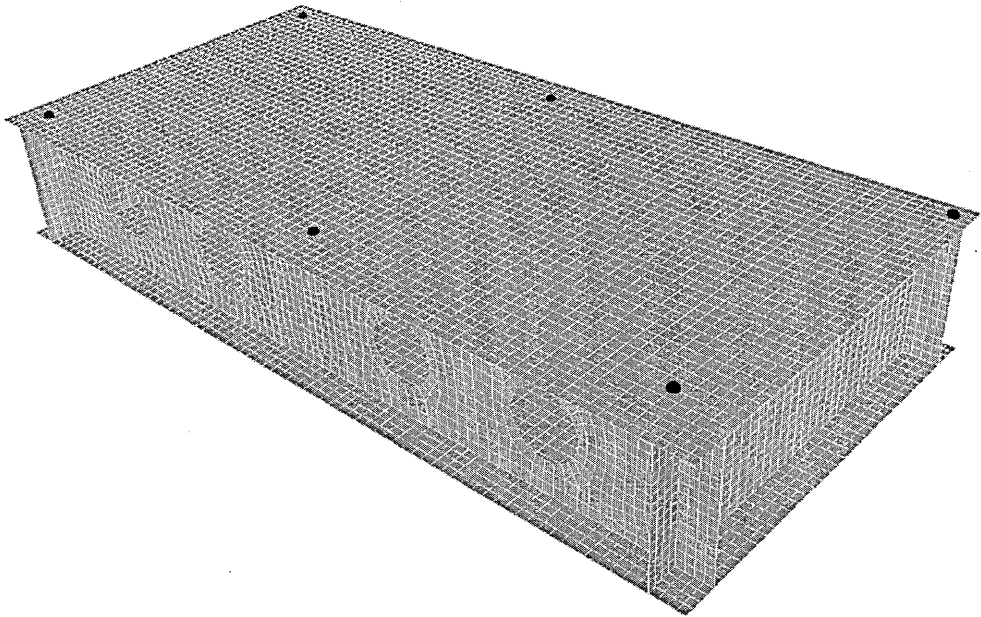


Figure 7.3: Positions where Time histories are recorded.

Filter	Initial Peak [g]	Constant Acceleration Level [g]	Constant Acceleration Duration [ms]
CFC60	49	40	15
CFC180	80	40	17
CFC600	122	40	17
CFC1000	136	40	17

Table 7.5: Acceleration Levels for Subfloor, Simulation 1.

Because of the constraints applied to the nodes with added ballast, the time histories of these nodes are identical (see Figure 7.4). They show an initial peak followed by a relatively constant acceleration level of 40g. This constant acceleration lasts around 17 ms. The magnitude and shape of the initial peak, on contact with the rigid wall, is highly dependent on the cut-off frequency used in the CFC filter (see Figure 7.5). For higher cut-off frequencies the peak becomes sharper and is much higher, the exact values are listed in Table 7.5. The time histories of the nodes near the central longeron show an initial peak on impact followed by a period of nearly 25ms of relatively low accelerations. This is because this part of the structure is not loaded by any ballast and hence comes to rest almost immediately after impact. The high oscillatory accelerations at the around 27ms is the moment where the subfloor rebounds from the rigid surface, as can be seen from the velocity time histories shown in Figure 7.6.

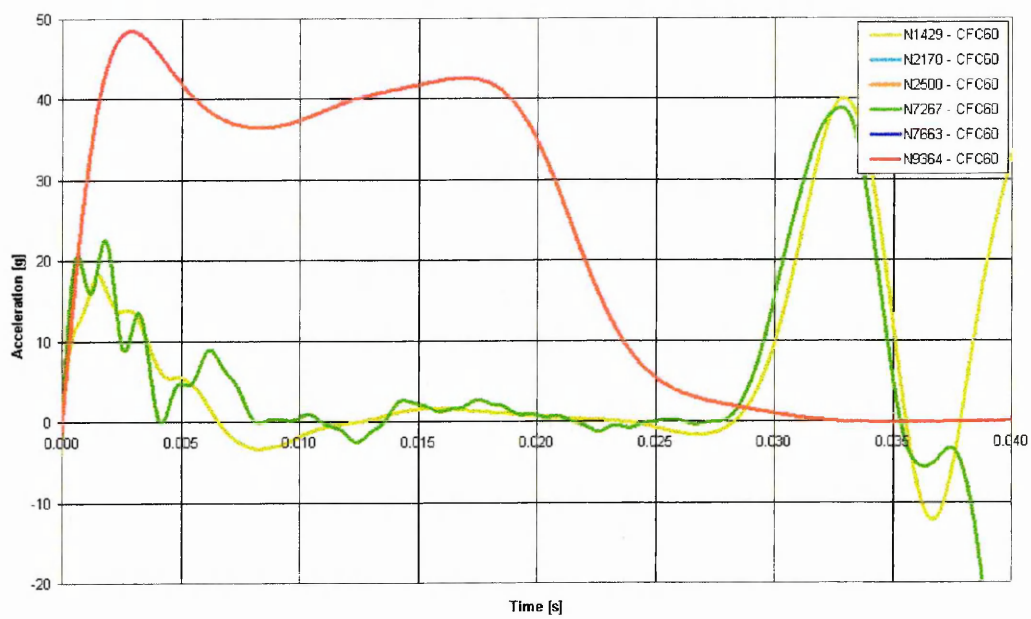


Figure 7.4: Time histories for Subfloor Impact, Simulation 1.

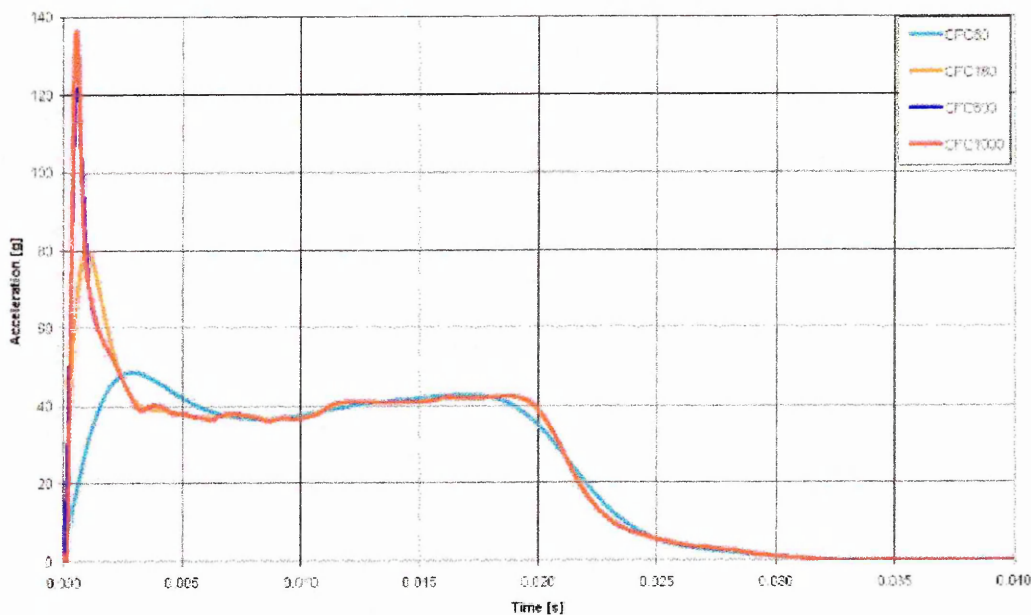


Figure 7.5: Influence of Filtering on Time histories for Subfloor, Simulation 1.

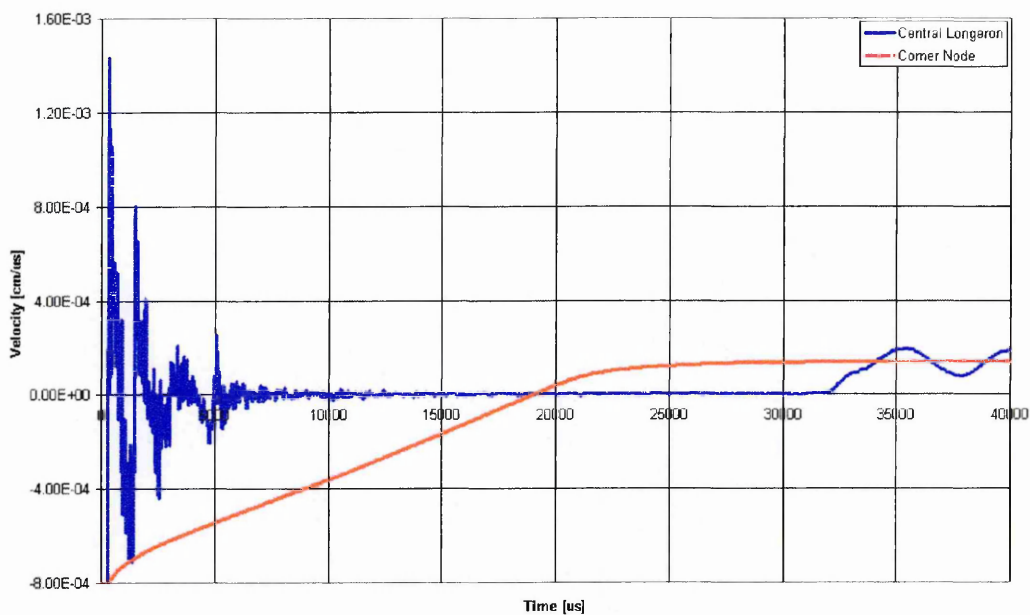


Figure 7.6: Velocity Time histories for Subfloor, Simulation 1.

Figure 7.7 and Figure 7.8 show stateplots of effective plastic strain contours after 40ms. The scale is from 0% (blue) to 10% (red). It is clear that the regions where plastic deformation occurs are directly underneath the ballast points. Some plastic deformation is also present in the frames and longerons in the regions of high curvature due to buckling. The buckled S-shape of the longeron is clearly visible in Figure 7.8.

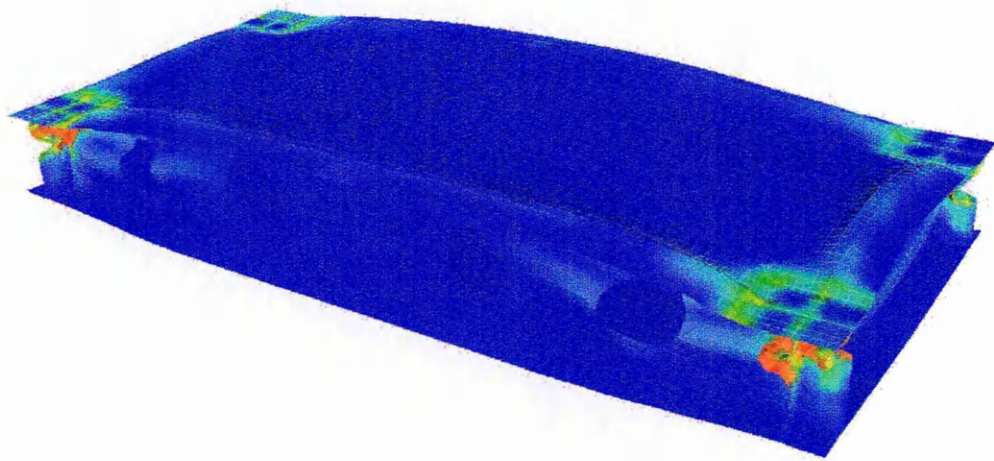


Figure 7.7: Effective Plastic Strain [0-10%] in subfloor at 40 ms, Simulation 1.

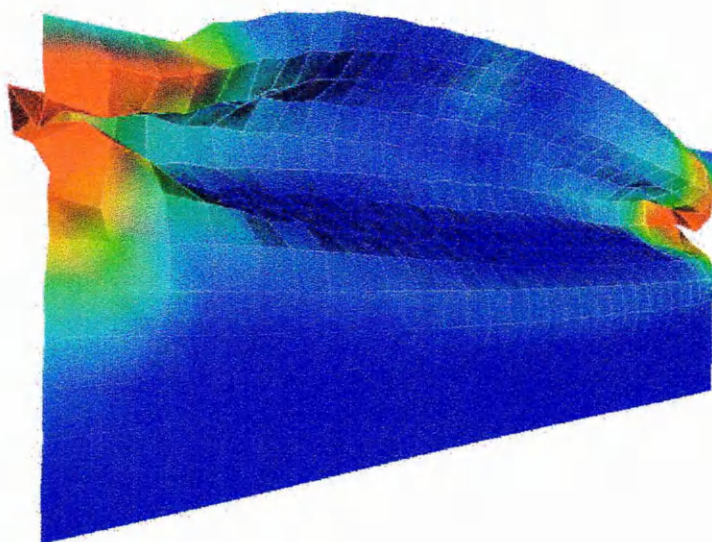


Figure 7.8: Effective Plastic Strain [0-0.1%] in Longeron at 40 ms, Simulation 1.

### 7.3.2 Anisotropic Material Model

The second simulation on rigid surface uses the anisotropic plastic material model which was described in Chapter 5. The material parameters used in this simulation were representative of the aluminium 2024-T3 alloy (see Table 7.6). No data was available on the ratio between isotropic and kinematic hardening. The assumption was made that there was no kinematic hardening. High speed tensile tests with up to strain rates of  $112\text{s}^{-1}$  performed by DLR [ 53 ] show that Al 2024 – T351 is not strain rate dependent. As a consequence no strain rate dependent material parameters are included in the material input data.

The time history results of this second simulation also show an initial peak followed by a period of around 19 ms of relatively constant acceleration. The level of acceleration is again around 40 g. The dependence on the cut-off frequency used in the filter of the magnitude and shape of the initial peak is shown in Figure 7.10, the values are listed in Table 7.7. The initial peaks are around 15 to 20% lower than the peaks obtained with the Von Mises plasticity model (cf. Table 7.5). The time histories of the nodes near the central longeron show an initial peak of around 22 g on impact followed by a period of nearly 25 ms of relatively low accelerations. The comparison of the CFC600 filtered time histories of simulation 1 and 2 are shown in Figure 7.11. The material anisotropy has a clear effect on the acceleration time histories. The initial acceleration peak is lower (108 g vs 122 g). These accelerations remain lower until around 10 ms after impact. The 40 g acceleration level also lasts 2 ms longer in comparison to the results obtained with the Von Mises plasticity material model. From Figure 7.12 and Figure 7.13 it can be seen that the location and levels of effective plastic strain and the mode of collapse of the longeron are also very similar.

Material Constant	Value
$E_x$ [GPa]	72.0
$E_y$ [GPa]	66.0
$E_z$ [GPa]	72.0
R	3.444
P	0.612
Q	1.0000
Q	0.8210
Q	0.6928
$v_{yx}$	0.33
$v_{zx}$	0.33
$v_{zy}$	0.33
Yield Stress X-dir [MPa]	360.0
Hardening Modulus [MPa]	667.0
$G_{yz}$ [GPa]	28.0
$G_{xy}$ [GPa]	28.0
$G_{xz}$ [GPa]	28.0
Density [g/cm <sup>3</sup> ]	2.71

Table 7.6: Anisotropic Material Properties for Al 2024.

Filter	Initial Peak [g]	Constant Acceleration Level [g]	Constant Acceleration Duration [ms]
CFC60	41	40	16
CFC180	68	40	19
CFC600	108	40	19
CFC1000	122	40	19

Table 7.7: Acceleration Levels for Subfloor, Simulation 2.



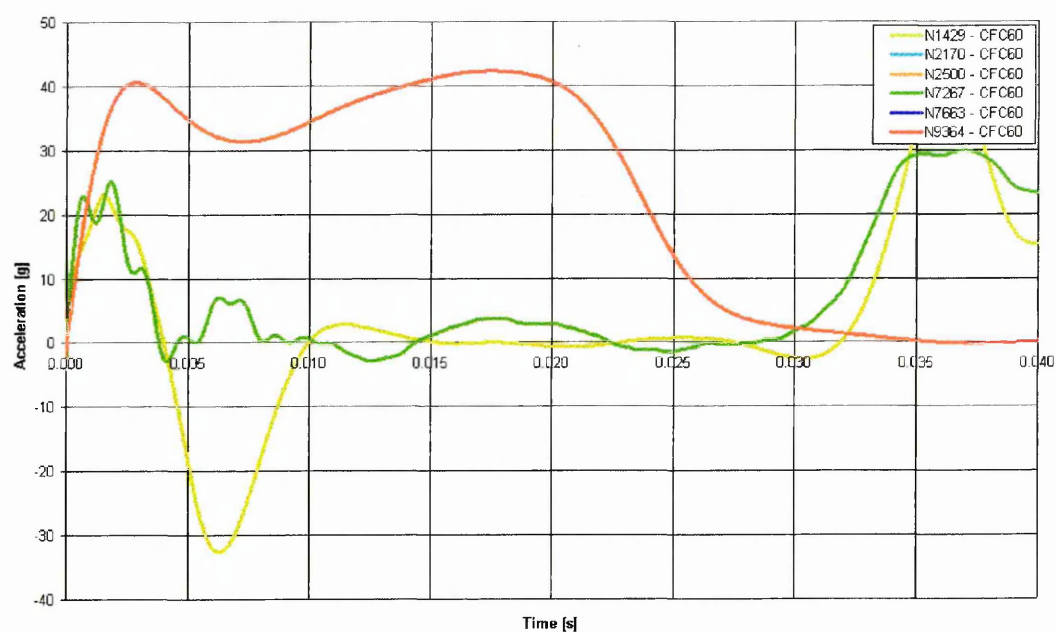


Figure 7.9: Time histories for Subfloor, Simulation 2.

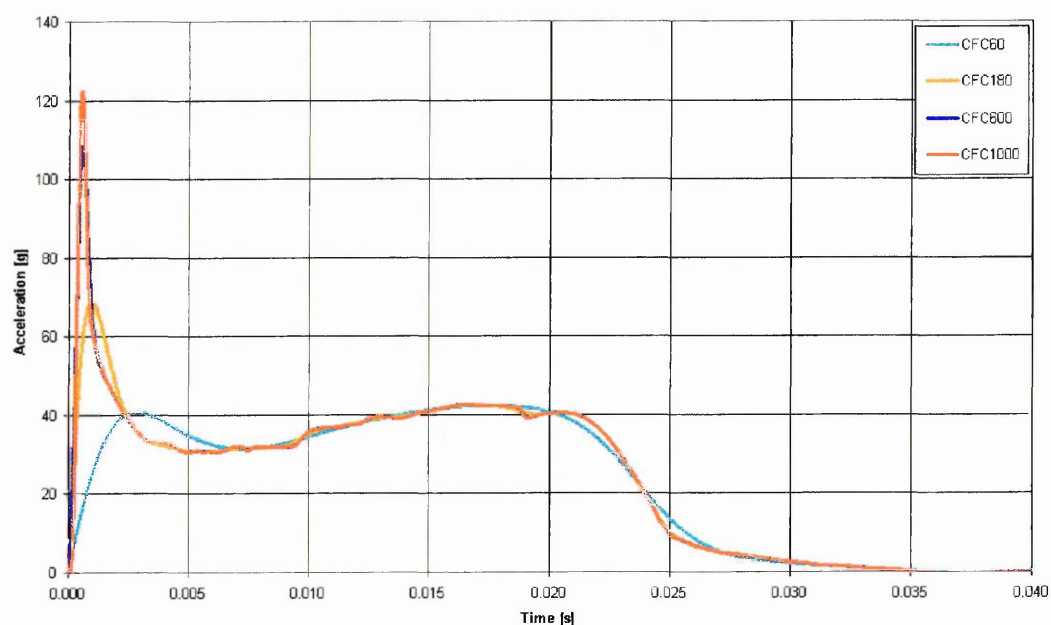


Figure 7.10: Influence of Filtering on Time histories for Subfloor, Simulation 2.



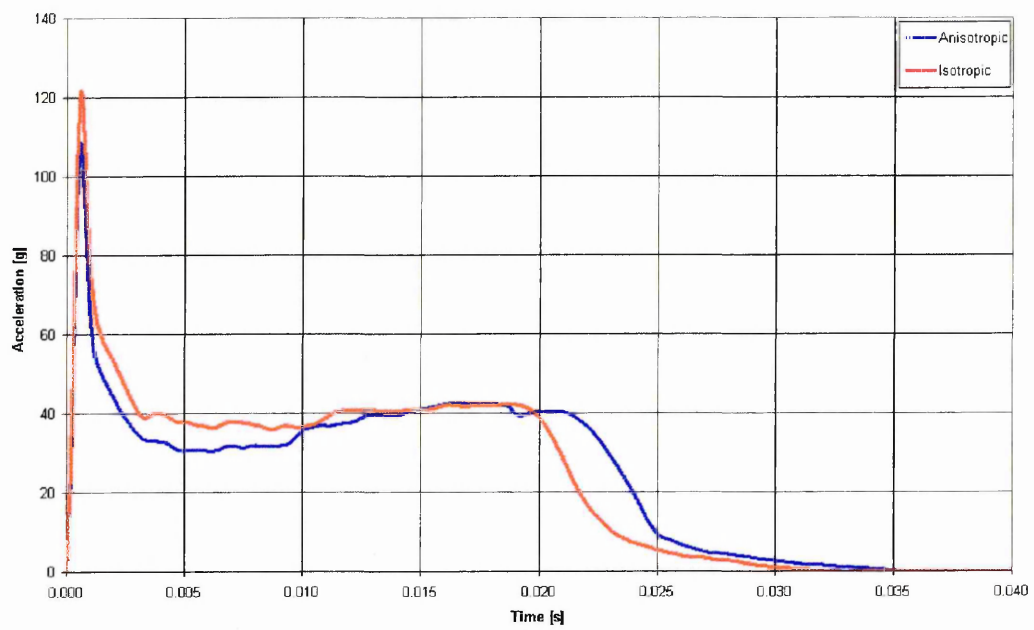


Figure 7.11: Comparison of CFC600 filtered Time histories for Simulation 1 & 2.

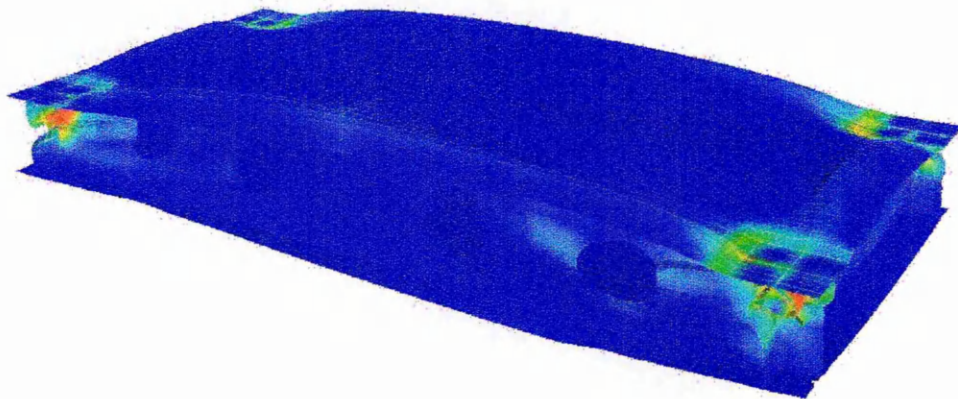


Figure 7.12: Effective Plastic Strain [0-10%] in Subfloor at 40 ms, Simulation 2.

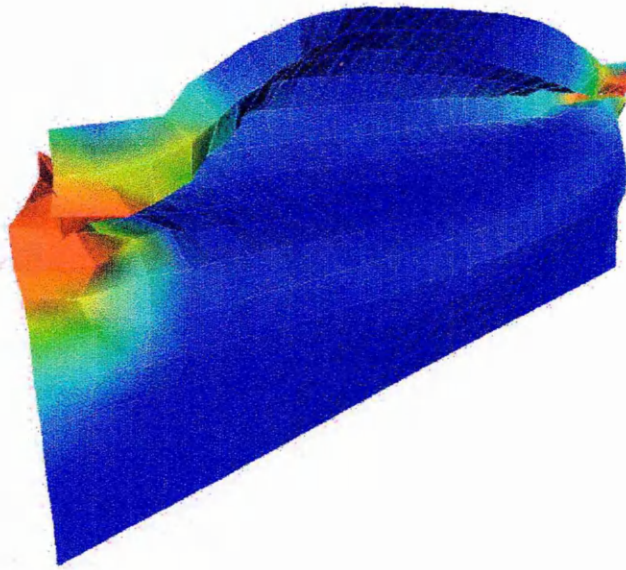


Figure 7.13: Effective Plastic Strain [0-10%] in Longeron at 40 ms, Simulation 2.

## 7.4 Simulation Results – Water Impact

### 7.4.1 isotropic

The two simulations of rigid impact have been repeated for the case of an impact on water. The model setup (see Figure 7.14) was identical for the subfloor. The only change is the substitution of the rigid wall with a volume of water ( $170 \times 100 \times 51 \text{ cm}^3$ ) discretised with SPH particles. A total of 31416 particles was used, the particle spacing being equal to 3.0 cm. As in the rigid shape impact simulations presented in the previous chapter, a fluid material model with Gruneisen equation of state has been used, with parameters defined in Table 6.2.

The acceleration time histories are plotted in Figure 7.15. The g-levels for the four corner nodes (nodes 2170, 2500, 7663 and 9364) are on left scale, while for the two nodes located near the central longeron of the subfloor the g-levels are on the right scale. The accelerations of the corner nodes are much lower than the equivalent rigid surface impact simulation, but show the same high peak on contact followed by a period of constant acceleration (see Figure 7.16). The initial peak is much lower, 18.2g compared to 122.0g (with CFC600 filtering). The constant level is around 12.2g while in the rigid surface impact 40g was reached. The acceleration levels for the nodes near the central longeron are completely different and show a high initial peak of 108 g.

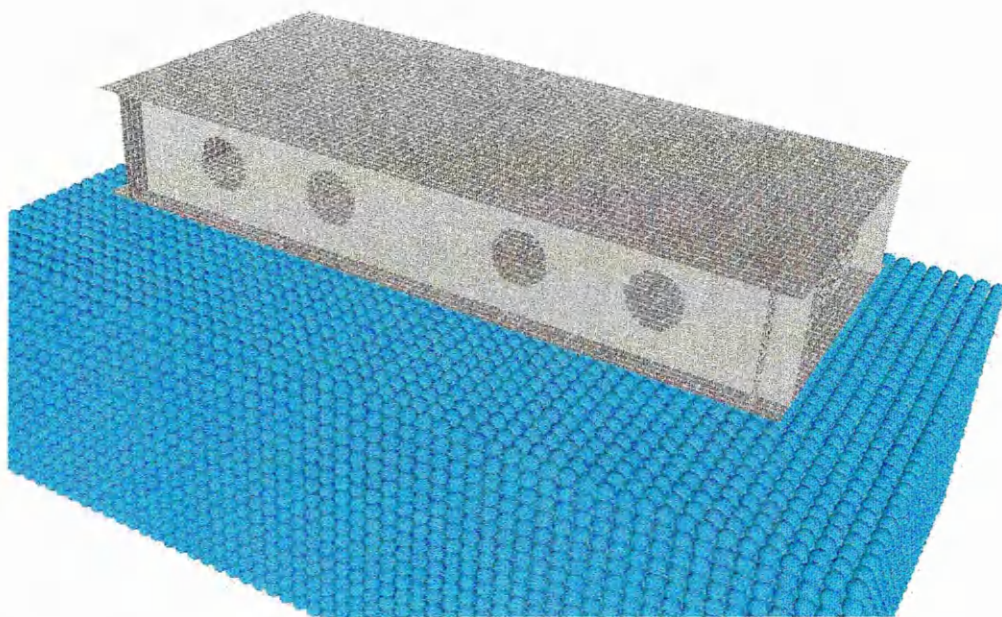


Figure 7.14: Initial Model Configuration, Simulation 3.

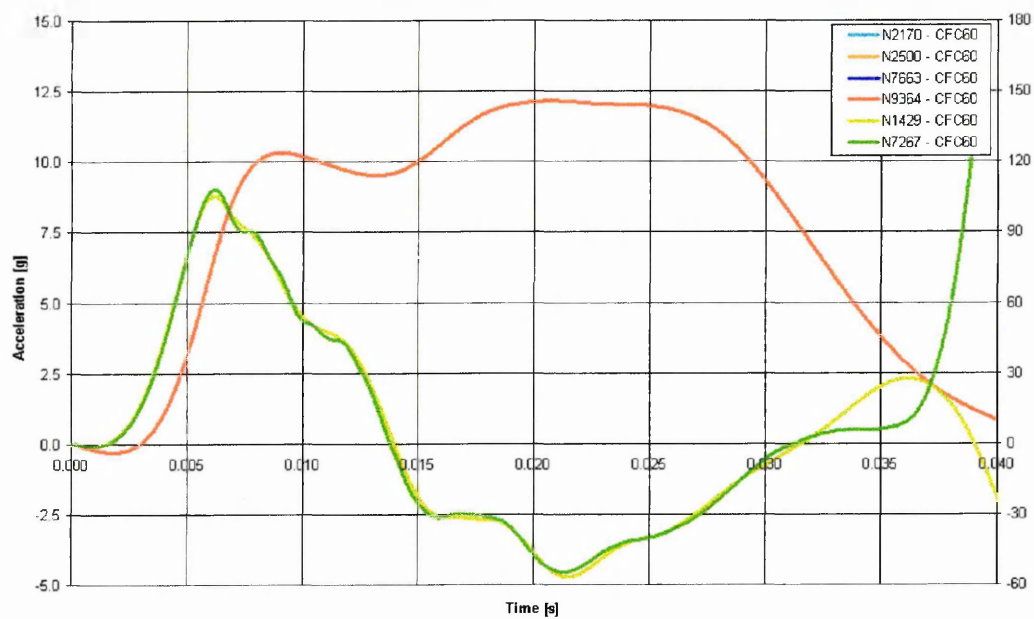


Figure 7.15: Time histories for Subfloor, Simulation 3.

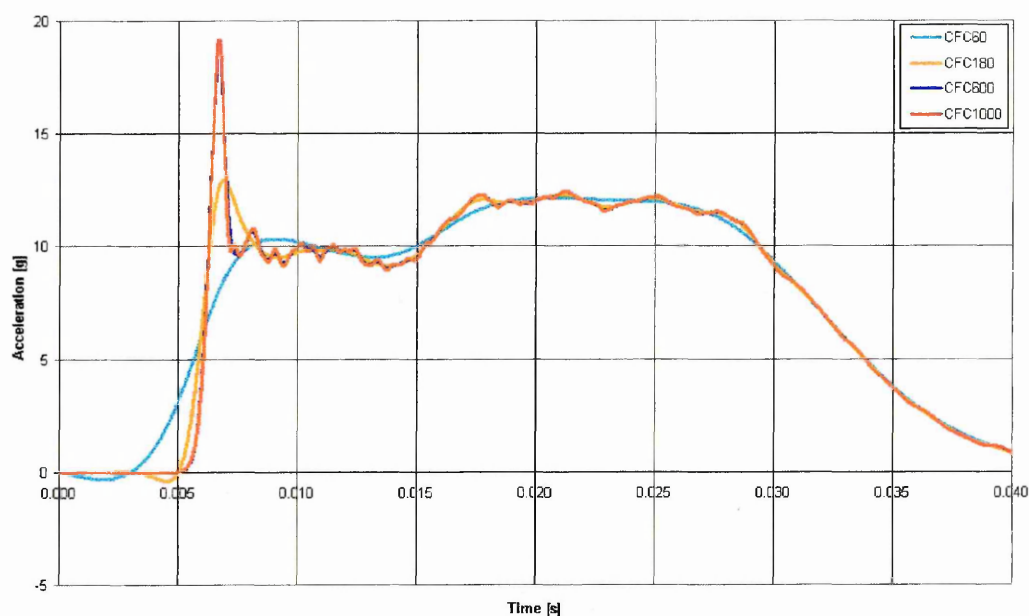


Figure 7.16: Influence of Filtering on Time histories for Subfloor, Simulation 3.

Filter	Initial Peak [g]	Constant Acceleration Level [g]	Constant Acceleration Duration [ms]
CFC60	10.3	12.2	-
CFC180	13.0	12.2	21
CFC600	18.2	12.2	22
CFC1000	19.1	12.2	22

Table 7.8: Acceleration Levels for Subfloor, Simulation 3.

The deformed shape after 40 ms is shown in Figure 7.17 to Figure 7.20. Figure 7.17 shows a cross section through the centre of the model in the transverse direction of the helicopter. This demonstrates that the FE-SPH coupling is able to deal with this kind fluid-structure interaction. The lower skin is deformed, and the overall bending of the subfloor due to the water load is clearly visible. The locations of high plastic deformation are different when compared to the rigid surface impact. While in the rigid surface impact the plastic deformation was located directly underneath the ballast locations, the water impact simulation shows three areas of plastic deformation. There is still some plastic deformation under the ballast points, but there also locations of high plastic deformation in the longerons and frames (see Figure 7.18 and Figure 7.19). The mode of collapse of the longerons and frames is also completely different.



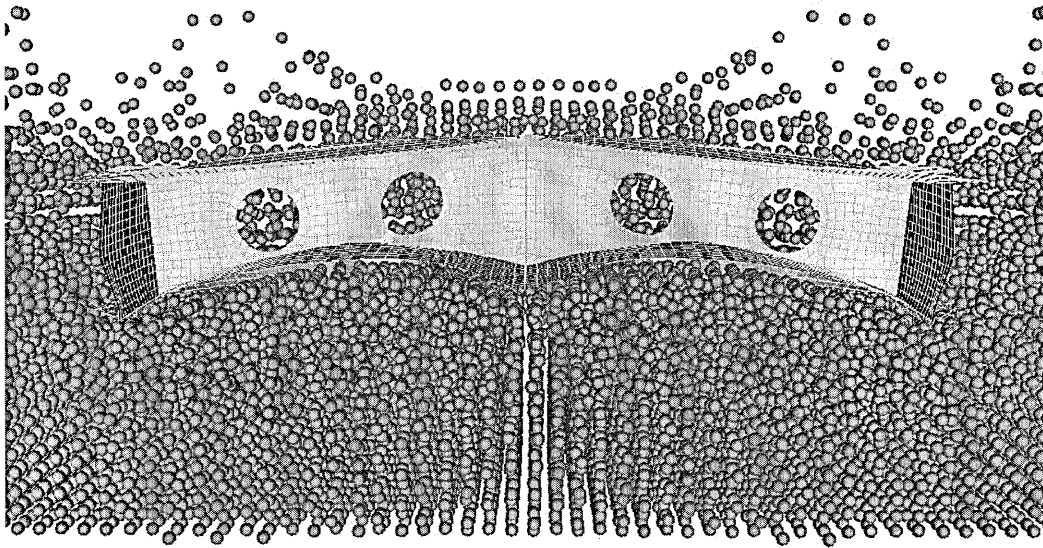


Figure 7.17: Deformed Shape after 40ms, Simulation 3.

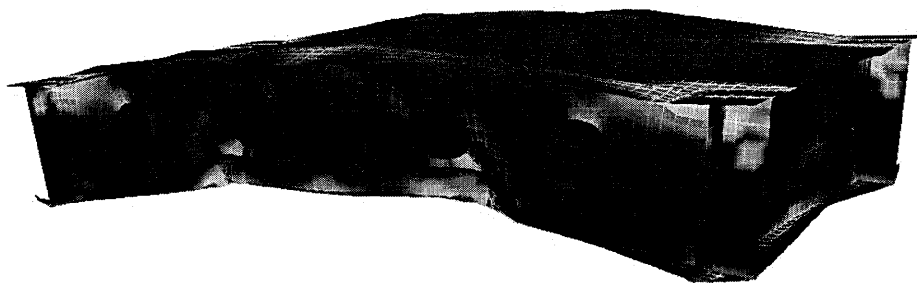


Figure 7.18: Effective Plastic Strain (0-10%) after 40ms, Simulation 3.

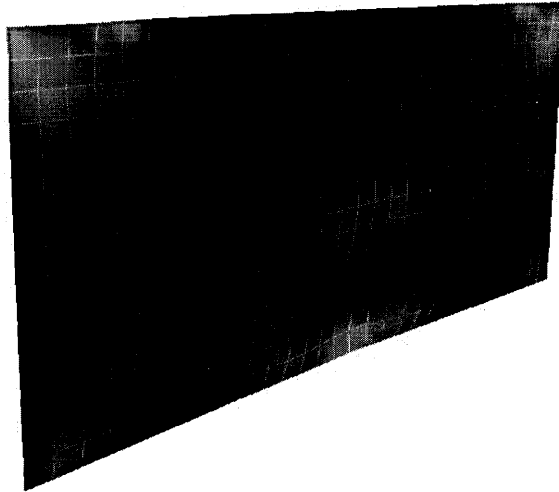


Figure 7.19: Mode of Collapse of Longerons, Simulation 3.

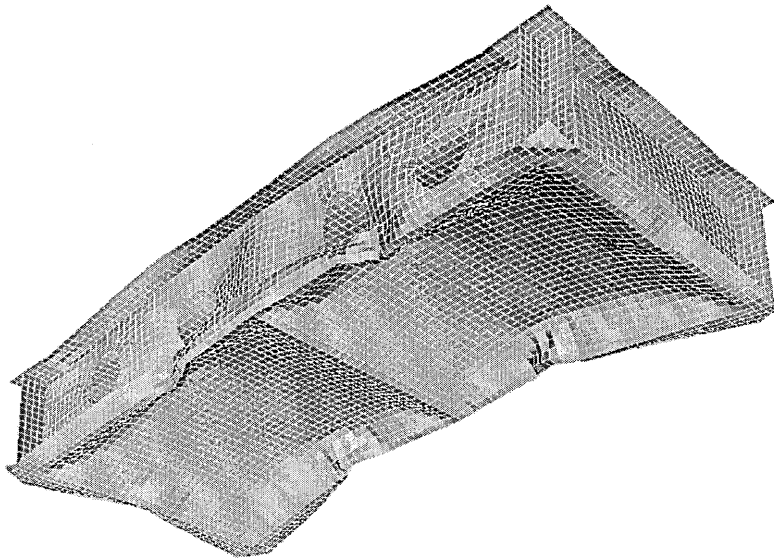


Figure 7.20: Lower Skin Deformation after 40ms, Simulation 3.

#### 7.4.2 Anisotropic

The acceleration time histories obtained with the anisotropic plasticity are plotted in Figure 7.21. The g-levels for the four corner nodes (nodes 2170, 2500, 7663 and 9364) are on left scale, while for the two nodes located near the central longeron of the subfloor the g-levels are on the right scale. The accelerations of the corner nodes are similar to simulation 3, and show the same peak on contact followed by a period of constant acceleration (see Figure 7.22). The initial peak is identical to simulation 3 at

18.2g with CFC600 filtering. This suggests that the magnitude of this peak is determined by the fluid-structure interaction (see Figure 7.23). After the initial peak the acceleration level is around 0.6 g (ie. 5%) lower than simulation 3. The constant acceleration level is 11.6g.

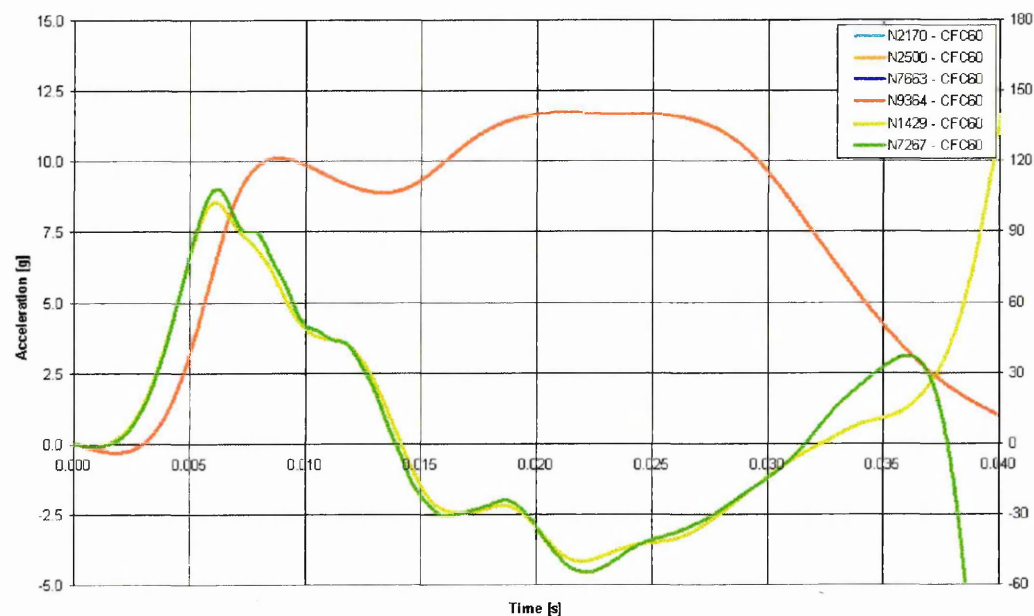


Figure 7.21: Time histories for Subfloor, Simulation 4.

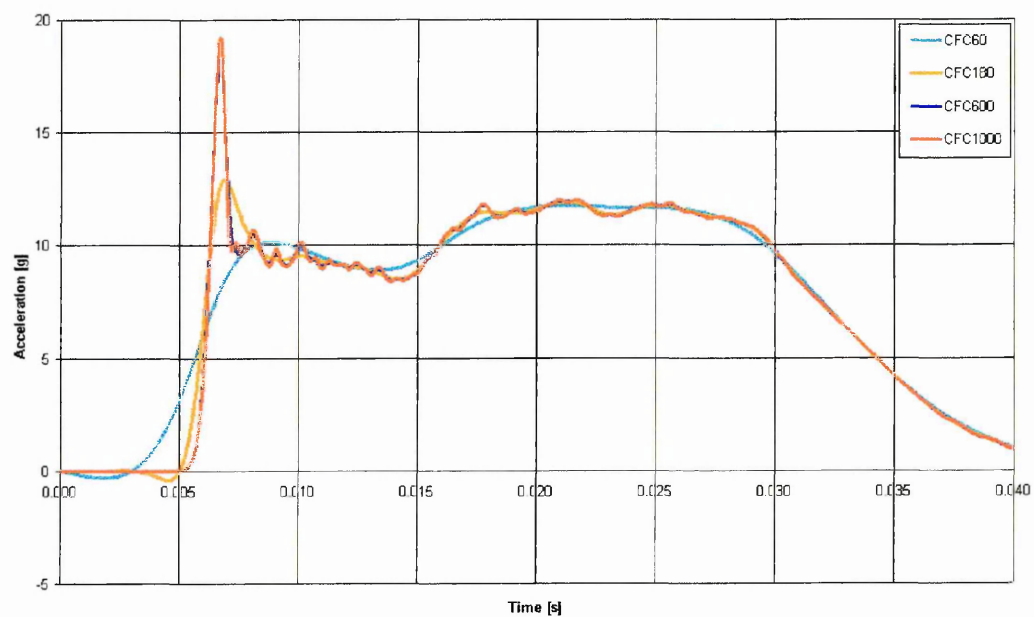


Figure 7.22: Influence of Filtering on Time histories for Subfloor, Simulation 4.

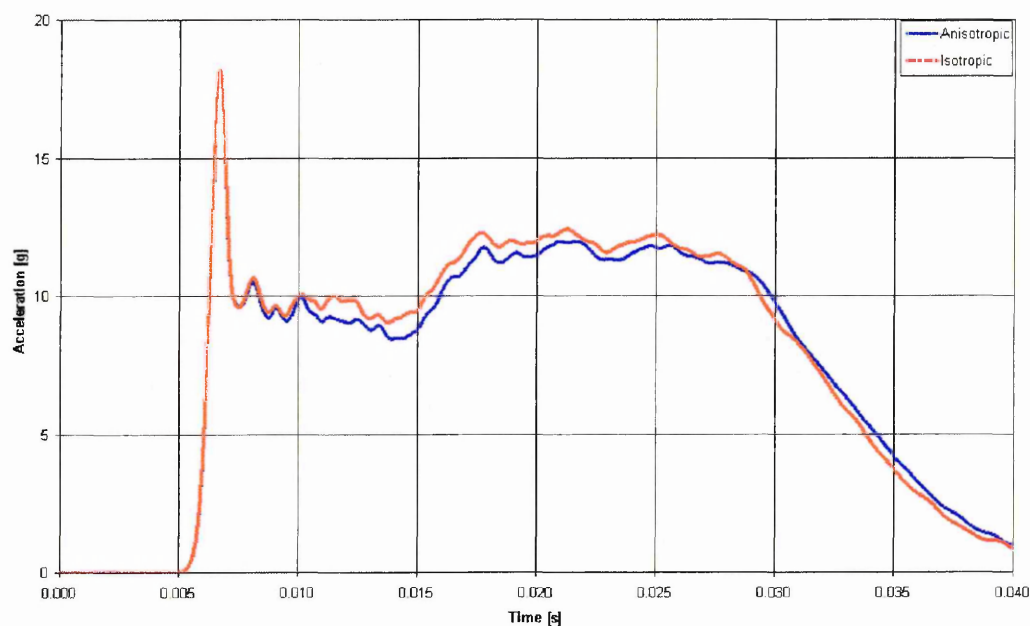


Figure 7.23: Comparison between Time histories of Simulation 3 & 4.

Filter	Initial Peak [g]	Constant Acceleration Level [g]	Constant Acceleration Duration [ms]
CFC60	10.3	11.6	-
CFC180	13.0	11.6	21
CFC600	18.2	11.6	22
CFC1000	19.1	11.6	22

Table 7.9: Acceleration Levels for Subfloor, Simulation 4.

The deformed shape after 40 ms is shown in Figure 7.24 to Figure 7.27. Figure 7.24 shows a cross section through the centre of the model in the transverse direction of the helicopter. The deformation, locations of plastic deformation and mode of collapse of the longerons and frames is very similar to simulation 3 using Von Mises plasticity.



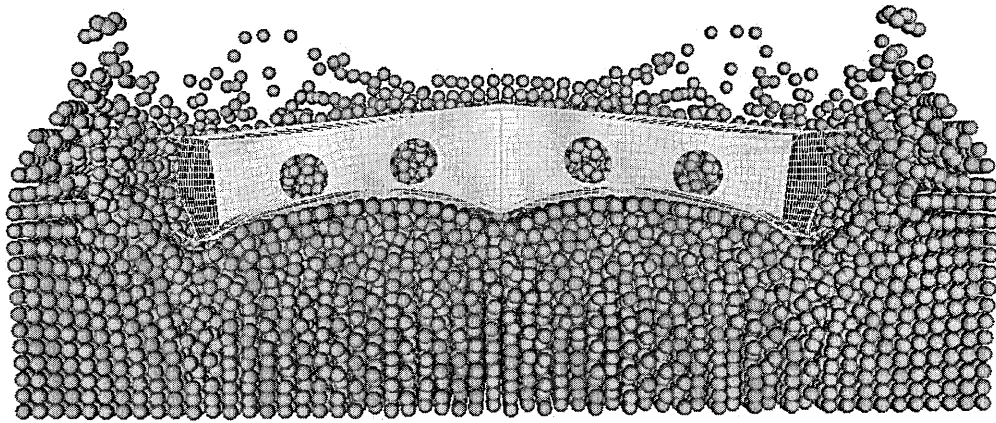


Figure 7.24: Deformed Shape at 40ms, Simulation 4.

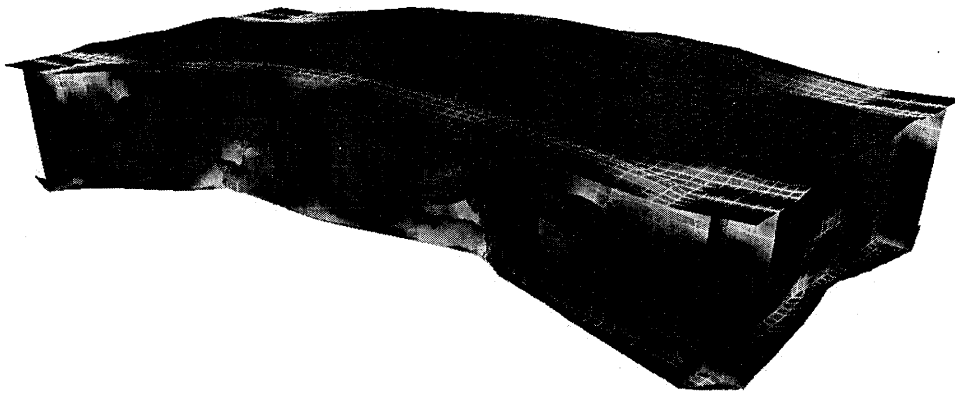


Figure 7.25: Effective Plastic Strain (0-10%) at 40ms, Simulation 4.

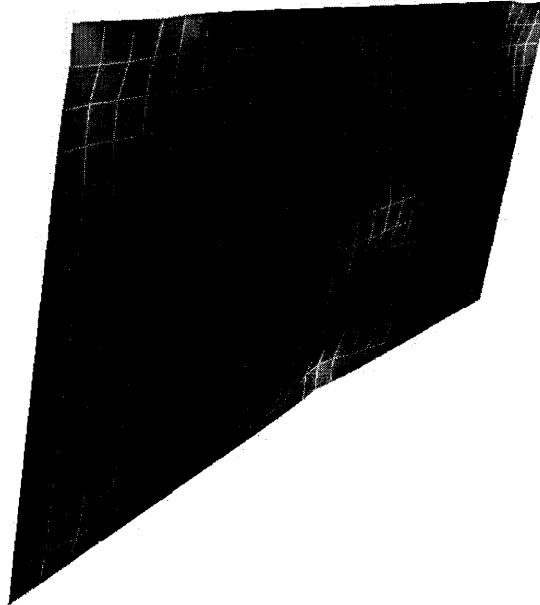


Figure 7.26: Mode of Collapse of Longerons, Simulation 4.

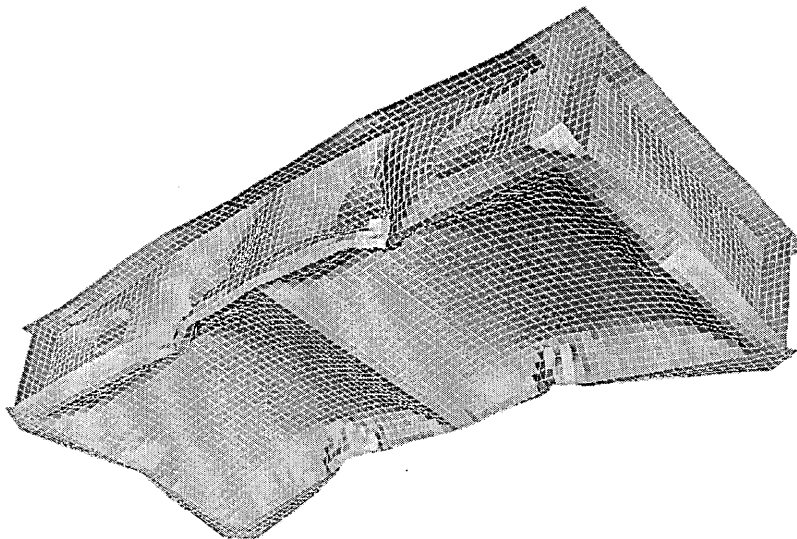


Figure 7.27: Lower Skin Deformation at 40ms, Simulation 4.

### 7.4.3 Skin Failure

To conclude this chapter a simulation with a thinner and locally weakened lower skin was performed. The purpose of this simulation was demonstrate the capability of the combined FE-SPH modelling technique to deal with skin failure and the associated water ingress, and the influence this may have on the acceleration levels and structural collapse. The thickness was reduced from 0.64 mm to 0.24 mm. Furthermore a failure strain was specified at 15% throughout the model, and in one corner a set of elements

was assigned a failure strain of 1%. An element failure criterion was added to the material model which removes the element from the calculation once it has reached its failure strain. First a simulation of the model with reduced skin thickness was performed without failure to obtain acceleration time histories for the model without skin failure. The acceleration time histories are shown in Figure 7.28. From this figure it is clear that a reduction of the lower skin thickness significantly affects the acceleration time histories. The initial peak is 26% lower at 13.5g compared to 18.2g. Up to 20ms after impact the accelerations with a 0.24mm skin remain lower. At 20ms they reach a similar magnitude than the accelerations with a 0.64mm skin (11.5g), after this point they steadily decrease. When the skin fails in one corner of the subfloor, a further reduction in acceleration levels is observed. The initial peak on impact remains identical, the first element fails at 8ms. All the weakened elements have failed after 12ms (see Figure 7.29). From that point onwards a clear difference in the magnitude of acceleration can be observed.

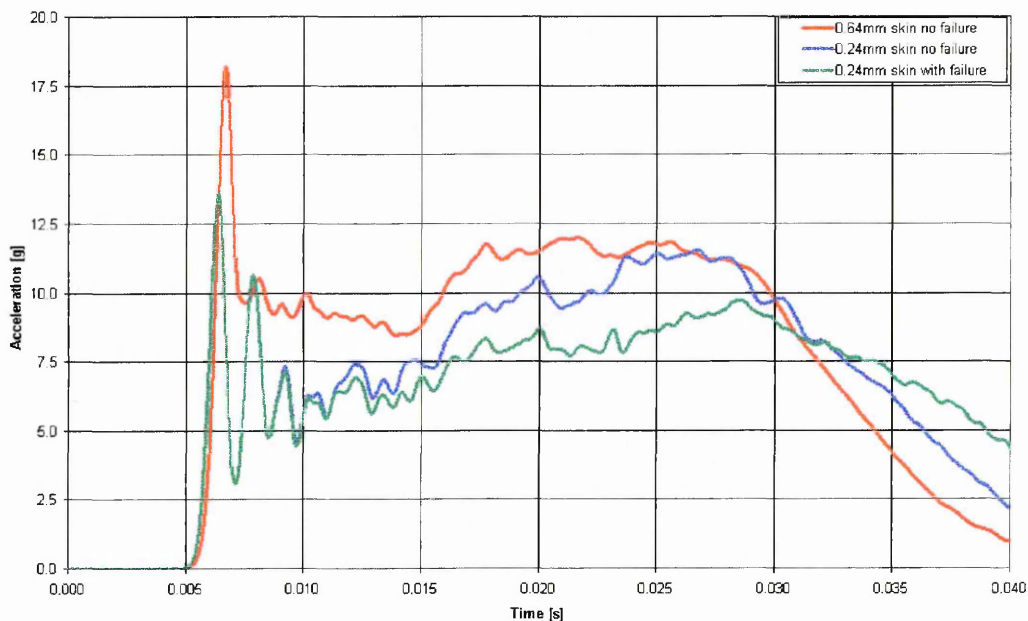


Figure 7.28: Overview of Time histories, Simulations 3, 5a, 5b.

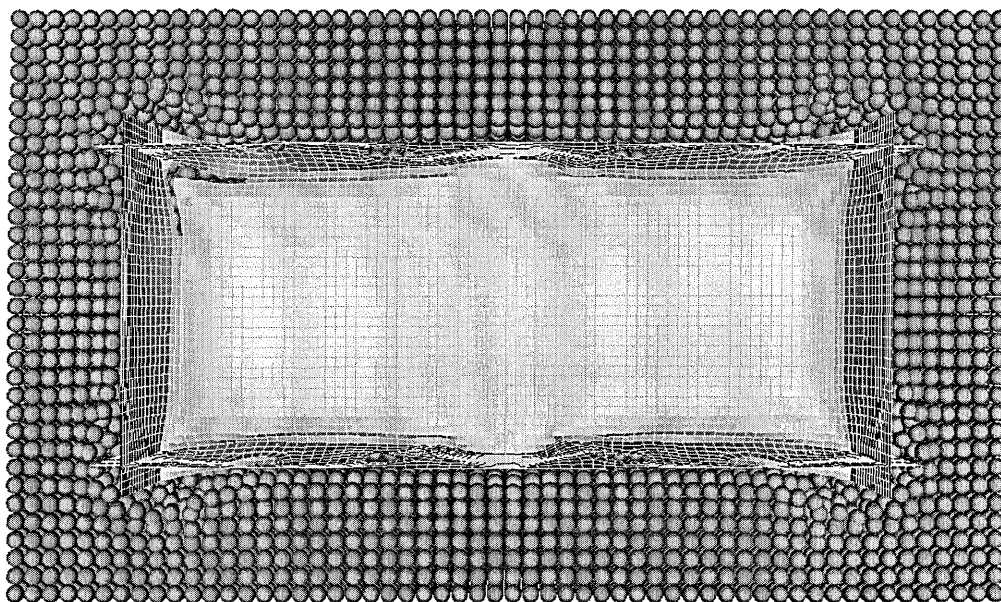


Figure 7.29: Failed Elements after 12 ms, Simulation 5b.

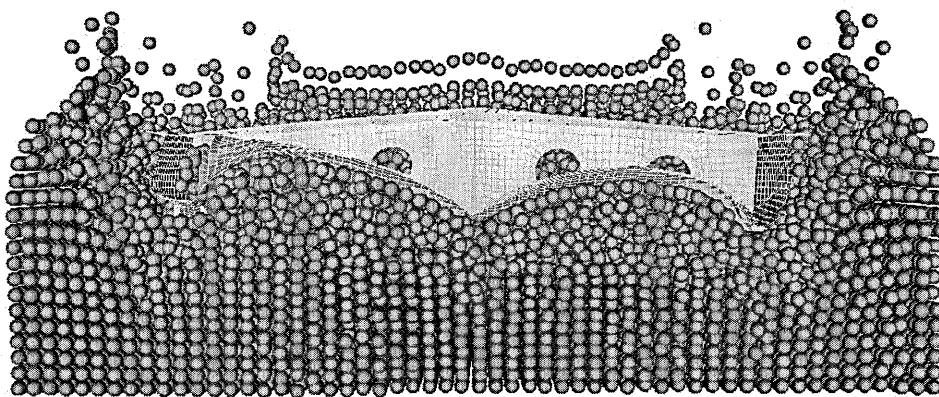


Figure 7.30: Cross Section after 40 ms, Simulation 5.

In Figure 7.30 to Figure 7.35 the deformed shape of the subfloor is shown. Due to the skin failure the structure collapses asymmetrically. From Figure 7.30, Figure 7.31 and Figure 7.32 it can be seen that the frames collapse in a different way than without skin failure. Due to the skin failure in one box, the frames on that side of the central longeron deform less. When the skin fails less load will be transferred into these frames, so they will absorb less energy. The deformation of the longeron and the global areas of high plastic deformation are shown in Figure 7.33 and Figure 7.34. Figure 7.35 shows the entry of water into the structure after 60 ms.

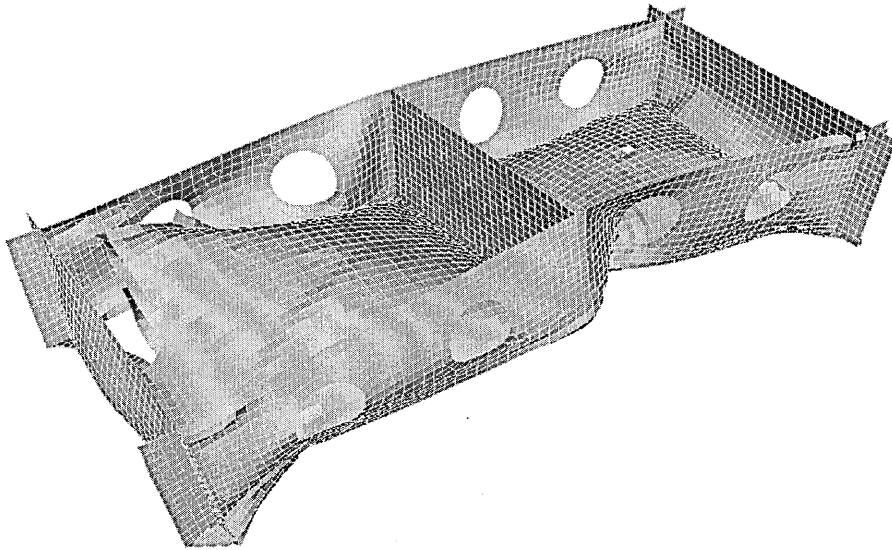


Figure 7.31: Isometric View after 40 ms, Simulation 5.

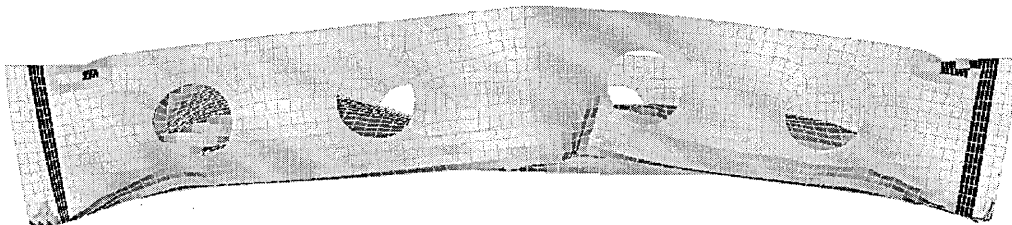


Figure 7.32: Side View of Subfloor after 40 ms, Simulation 5.

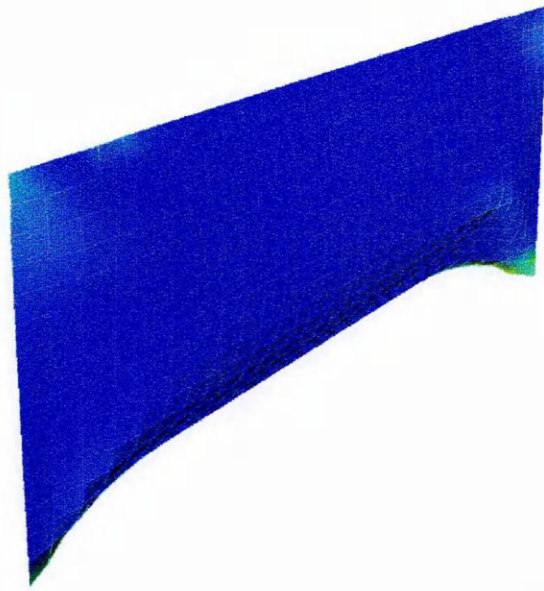


Figure 7.33: Longeron Deformation after 40 ms, Simulation 5.

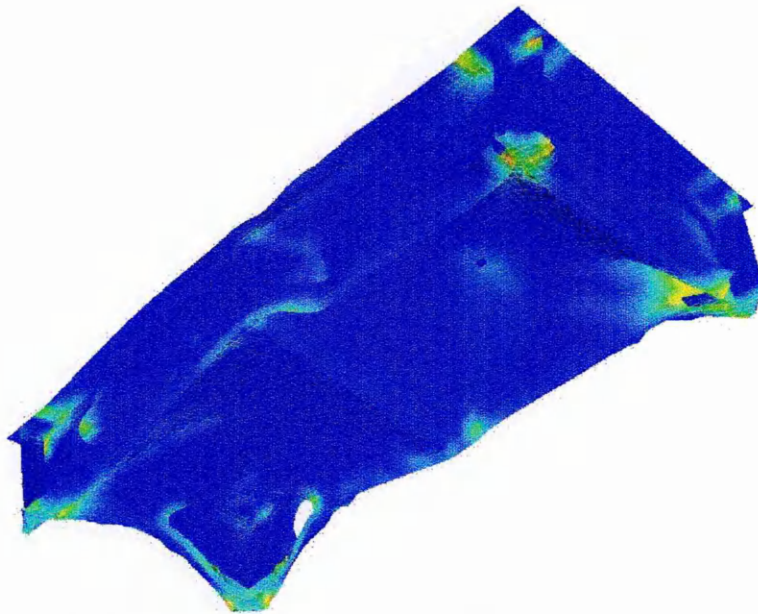


Figure 7.34: Effective Plastic Strain [0 – 10%] after 40 ms, Simulation 5.



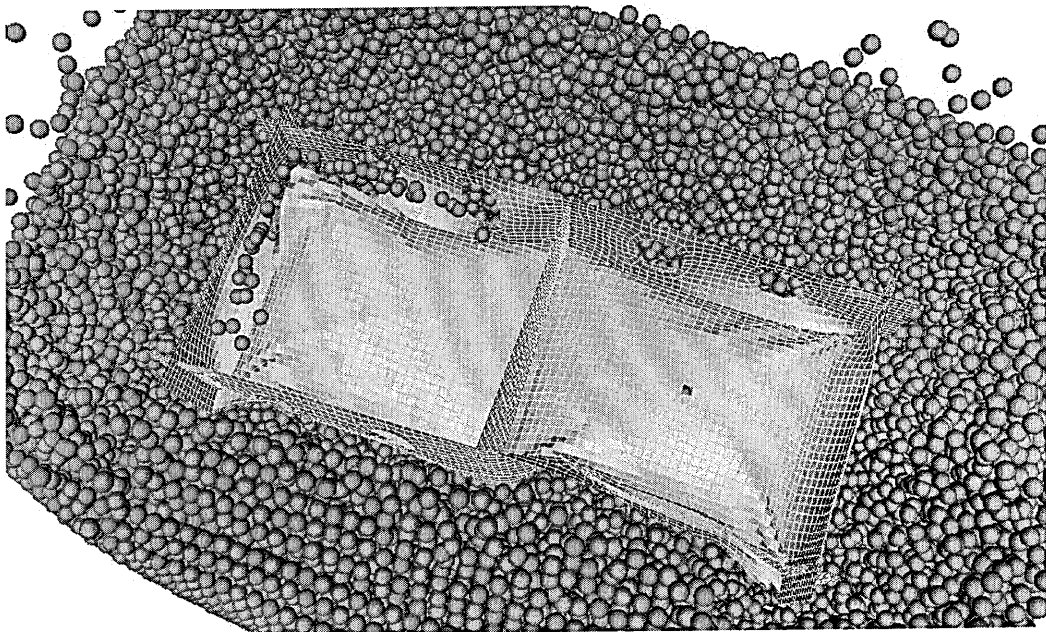


Figure 7.35: Skin Failure after 60 ms, Simulation 5.

## Chapter 8 1D Lagrangian SPH

### 8.1 Lagrangian Kernel

One of the major problems in the SPH method is the presence of a tensile instability in the formulation of solid mechanics problems. This instability was identified by Swegle [ 83 ], and has also been discussed by Belytschko [ 8 ], [ 9 ], Bonet [ 15 ], [ 16 ], and Randles, Petschek, Libersky and Dyka [ 76 ]. This instability is present when so-called Eulerian kernels are used. The term Eulerian kernel is used because the kernel values and derivatives are calculated using the current positions of the particles. This means that particle positions used to calculate kernel values, change from timestep to timestep, particles may even enter or leave the kernel sums. This is of course a very attractive feature when modelling large deformation problems.

A solution to the problem of tensile instability is the use of a Lagrangian kernel. This means that the kernel values and derivatives are calculated with a fixed reference particle distribution. In doing so one obviously partly sacrifices the main advantage of SPH, the ability to model extreme material distortions. However the use of a Lagrangian kernel would still allow to model crack propagation problems.

In order to derive the equations with a Lagrangian kernel one can start by formulating the conservation in a Lagrangian framework, ie. in a fixed reference configuration. The coordinates of a point in the reference configuration will be denoted in capitals as  $X$ , whereas the current coordinates will be written in lower case as  $x$ . If the momentum equation is formulated in terms of the nominal stress  $P$  the conservation of momentum equation is [ 7 ]:

$$\nabla \cdot P + b = \rho_0 a \quad \text{in } \Omega_0 \quad (\text{Eq. 8.1})$$

with boundary conditions:

$$P \cdot n_0 = \bar{t}_0 \quad \text{on } \Gamma_{t0} \quad (\text{Eq. 8.2a})$$

$$P \cdot n_0 = 0 \quad \text{on } \Gamma_{to0} \quad (\text{Eq. 8.2b})$$

$$u = \bar{u} \quad \text{on } \Gamma_{u0} \quad (\text{Eq. 8.3})$$

where  $\Gamma_{t0} \cup \Gamma_{to0} \cup \Gamma_{u0} = \Gamma_0$ , and  $\Gamma_{t0} \cap \Gamma_{to0} = \emptyset$ ,  $\Gamma_{to0} \cap \Gamma_{u0} = \emptyset$ ,  $\Gamma_{t0} \cap \Gamma_{u0} = \emptyset$ .

The nominal stress is the transpose of the first Piola-Kirchhoff stress. By using this form of the stress tensor the momentum equation looks very similar to the momentum equation expressed in the Eulerian description. One simply replaces the Cauchy stress by the nominal stress and the density by the initial density. The nominal stress tensor is not symmetric, and is not very useful when expressing constitutive equations [ 7 ], but this issue will be addressed in the next section 8.2



The first step is the derivation of a weak form of the conservation of momentum equation. In order to do this the equation is multiplied by a test function  $\delta w$  and integrated over the domain of the problem  $\Omega_0$ . Note that the integration is over the reference domain  $\Omega_0$  not over the current domain  $\Omega$  as was the case in Chapter 2. This results in:

$$\int_{\Omega_0} \delta w \nabla \cdot \mathbf{P} \, dV = \int_{\Omega_0} \delta w \rho_0 a \, dV \quad (\text{Eq. 8.4})$$

Using the integration by parts and using Green's theorem and writing the acceleration as the second time derivative of the displacement,  $\ddot{u}$ :

$$\int_{\Gamma_{t_0} \cup \Gamma_{u_0}} (\delta w \mathbf{P}) \cdot \mathbf{n}_0 \, d\Gamma - \int_{\Omega_0} \nabla \delta w \cdot \mathbf{P} \, dV = \int_{\Omega_0} \delta w \rho_0 \ddot{u} \, dV, \quad (\text{Eq. 8.5})$$

or in Voigt notation:

$$\int_{\Omega_0} \rho_0 \{\delta w\}^t \{\ddot{u}\} \, dV + \int_{\Omega_0} \{\nabla \delta w\}^t \{\mathbf{P}\} \, dV - \int_{\Gamma_{t_0} \cup \Gamma_{u_0}} \{\{\delta w\}^t \{\mathbf{P}\}\} \cdot \mathbf{n} \, d\Gamma = 0 \quad (\text{Eq. 8.6})$$

This equation can be discretised in the space domain by using shape functions for both, the displacement trial functions and the test functions:

$$\{\delta w\} = [\mathbf{N}] \{\delta d\} \quad (\text{Eq. 8.7})$$

$$\{u\} = [\mathbf{N}] \{d\} = \sum \mathbf{N} d, \quad (\text{Eq. 8.8})$$

where  $[\mathbf{N}]$  is an array of shape functions, and  $\{d\}$  is the nodal displacement vector. The  $[\mathbf{N}]$  array in this case contains shape functions that are a function of the reference coordinates  $\mathbf{X}$ .

$$N_j(\mathbf{X}) = \frac{m_j}{\rho_{0j}} W(\mathbf{X} - \mathbf{X}_j) \quad (\text{Eq. 8.9})$$

Using these discretisation the Galerkin expression becomes:

$$\int_{\Omega_0} \{\delta d\}^T \rho_0 [\mathbf{N}]^T [\mathbf{N}] \{\ddot{d}\} \, dV + \int_{\Omega_0} \{\delta d\}^T [\mathbf{B}]^T \{\mathbf{P}\} \, dV = 0 \quad (\text{Eq. 8.10})$$

This equation can now be transformed in the familiar form by noting that  $\delta d$  and  $d$  are nodal values and hence can be taken out of the integral sign, resulting in:

$$\{\delta d\}^T \left( \int_{\Omega_0} \rho_0 [N]^T [N] dV \{\ddot{d}\} + \int_{\Omega_0} [B]^T \{P\} dV \{d\} \right) = 0 \quad (\text{Eq. 8.11})$$

This relation should hold for an arbitrary test function  $\delta d$ , so one obtains:

$$[M] \{\ddot{d}\} + \{f_i\} = 0 \quad (\text{Eq. 8.12a})$$

with

$$\{f_i\} = \int_{\Omega_0} [B]^T \{P\} dV \{d\}$$

and

$$[M] = \int_{\Omega} \rho [N]^T [N] dV \quad (\text{Eq. 8.13})$$

If one first rewrites the momentum equation as:

$$a = \frac{1}{\rho_0} \nabla \cdot P = \nabla \cdot \frac{\sigma}{\rho_0} + \frac{\sigma}{\rho_0^2} \cdot \nabla \rho_0$$

this results in:

$$\int_{\Omega_0} \delta w \ddot{u} dV = - \int_{\Omega_0} \nabla \delta w \cdot \left( \frac{P}{\rho_0} \right) dV - \frac{P_i}{\rho_{0i}} \cdot \int_{\Omega_0} \nabla \delta w dV$$

This then gives the final relation for the momentum equation in Voigt notation:

$$\int_{\Omega_0} [N]^T [N] dV \{\ddot{u}\} = - \int_{\Omega_0} [B]^T \cdot \frac{P}{\rho_0} dV - \int_{\Omega_0} \rho_0 [B]^T dV \cdot \frac{P_i}{\rho_{0i}} \quad (\text{Eq. 8.14})$$

So far these are formally identical to the equations in Chapter 2. The only differences are that the nominal stress is used instead of the Cauchy stress, the integrals are over the initial domain, and the shape functions use the initial particle positions.

## 8.2 1D Lagrangian SPH code

The semidiscretised equations obtained in the previous paragraph can be discretised in time using a central difference algorithm. In Eq. 8.14 one can again use a lumped matrix on the LHS of the equation. Because the density was on the RHS of the equation at the beginning of the derivation, the matrix is in fact a volume matrix. Therefore the

acceleration vector will be premultiplied by a ‘lumped volume’. This lumped volume can for example be the particle mass divided by its initial density:

$$\frac{m_i}{\rho_{0i}}$$

The initial density is used here because the integrals are over the initial domain  $\Omega_0$ . The central difference time integration loop using these equations is summarised in Figure 8.1.

Start of step  $n$

Know  $x^n, v^{n-1/2}, t^n, \Delta t^{n-1/2}, \sigma^{n-1}, \rho^n$

Store  $\Delta t^{n-1/2}$

Calculate strain rate using positions at half step:

$$x^{n-1/2} = x^n - \frac{1}{2} v^{n-1/2} \Delta t^{n-1/2}$$

$$L = \dot{F} F^{-1}$$

$$\dot{\epsilon}^{n-1/2} = \frac{1}{2} (L + L^T)$$

Update Density:

$$\rho^n = \rho^{n-1} + \dot{\rho}^{n-1/2} \Delta t^{n-1/2} = \rho^{n-1} + \nabla \cdot v^{n-1/2} \Delta t^{n-1/2}$$

Calculate Stress using Constitutive Model:

$$\dot{\sigma}^{n-1/2} = f(\dot{\epsilon}^{n-1/2})$$

$$\sigma^n = \sigma^{n-1} + \dot{\sigma}^{n-1/2} \Delta t^{n-1/2}$$

$$P^n = J F^{-1} \sigma^n$$

Solve Momentum equation (where  $f_c^n$  is the vector containing contact forces):

$$f_{int}^n = -\rho_{0i} \left( \int_{\Omega_0} [B]^T \cdot \frac{P}{\rho_0} dV + \int_{\Omega_0} [B]^T dV \cdot \frac{P_i}{\rho_{0i}} \right)$$

$$a^n = -M^{-1} (f_{int}^n + f_c^n)$$

Update time,  $\Delta t$  using scale factor for stability:

$$\Delta t^{n+1/2} = \text{scale factor} \times \Delta t_{crit}^{n+1/2}$$

$$t^{n+1} = t^n + \Delta t^{n+1/2}$$

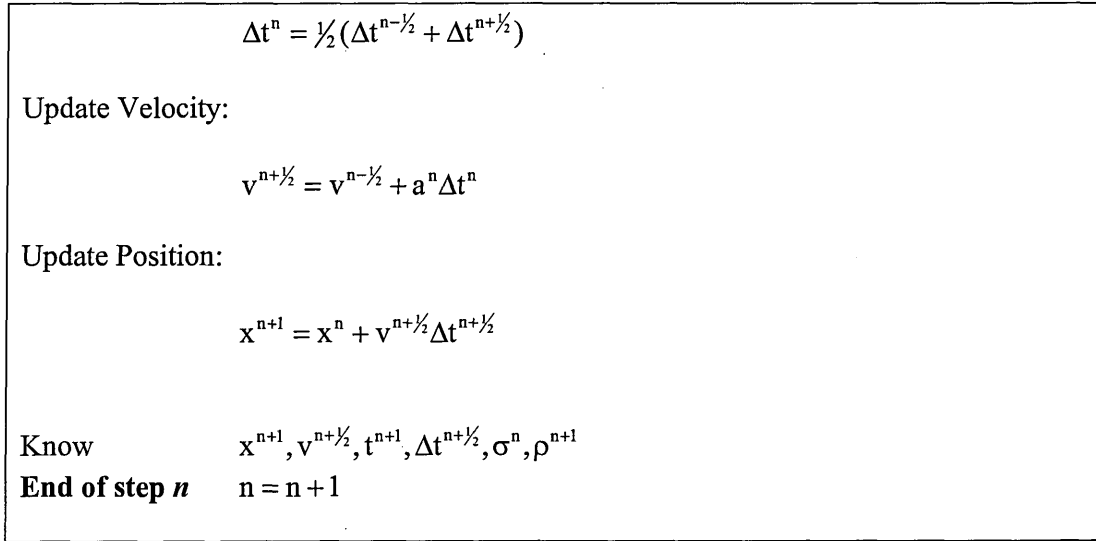


Figure 8.1 Central Difference Time Integration Loop for Lagrangian Kernel SPH.

One can see that the changes to the algorithm are in the calculation of the strain rate and the update of the momentum equation. The algorithm still uses the strain-rate tensor and the Cauchy stress tensor in the constitutive relations. One could also formulate a constitutive relation in terms of the nominal stress, but because it is not symmetric this is not very convenient [ 7 ]. Because the constitutive relations are formulated in terms of the strain-rate  $\dot{\boldsymbol{\epsilon}}$  and Cauchy stress  $\boldsymbol{\sigma}$ , and the deformation tensor  $\mathbf{F}$  and nominal stress  $\mathbf{P}$  (which are conjugate in power) are used in the Galerkin derivation, one will need to transform the deformation into strain rate. This can be done by using the following formula [ 7 ]:

$$\mathbf{L} = \dot{\mathbf{F}}\mathbf{F}^{-1} \quad (\text{Eq. 8.15})$$

$$\dot{\boldsymbol{\epsilon}}^{n-\frac{1}{2}} = \frac{1}{2}(\mathbf{L} + \mathbf{L}^T) \quad (\text{Eq. 8.16})$$

Once the Cauchy stress is obtained from the strain rate and the constitutive relationship, one has to transform the Cauchy stress into the nominal stress by using the following formula [ 7 ]:

$$\mathbf{P}^n = \mathbf{J}\mathbf{F}^{-1}\boldsymbol{\sigma}^n. \quad (\text{Eq. 8.17})$$

The nominal stress can then be used in the update of the momentum equation.

One of the more difficult problems with the implementation of a Lagrangian kernel SPH solver is that the material rotations with respect to the original configuration have to be calculated accurately. This is a consequence of using a total Lagrangian formulation (integrals and derivatives are taken with particles positions in the reference configuration), and the use of the Cauchy stress tensor (which is expressed in material coordinates, ie. current positions). This means that one has to either use a corotational Cauchy stress, or an objective stress rate (for example a Jaumann or Truesdell rate).

The use of a corotational Cauchy stress requires knowledge of the material rotation matrix  $R$ , which can be obtained by using polar decomposition of the deformation tensor:

$$F = UR,$$

where  $U$  is the stretch tensor. On the other hand if an objective rate is used then one needs to know the spin rate  $W$ .

In the case of a 1D Lagrangian kernel SPH code there are no material rotations, hence the implementation of a Lagrangian kernel SPH solver is greatly simplified. First of all the shape functions have to be calculated by using the reference coordinates and initial density:

$$N_j(X_i) = \frac{m_j}{\rho_{0j}} W(X_i - X_j)$$

Secondly one can approximate the deformation tensor  $F$ :

$$\langle F \rangle = \left\langle \frac{\partial x}{\partial X} \right\rangle = \sum_j \frac{m_j}{\rho_{0j}} x_j \nabla W(X_i - X_j)$$

Where the current position  $x_j$  is evaluated at time  $t_{n-1/2}$ . Because this a scalar value  $F^{-1}$  is simply the inverse of this value. Furthermore one can calculate the rate of deformation  $\dot{F}$ :

$$\langle \dot{F} \rangle = \sum_j \frac{m_j}{\rho_{0j}} v_j \nabla W(X_i - X_j)$$

The velocity gradient  $L$  is in the 1D case a simple multiplication of the two scalar values of  $\dot{F}$  and  $F^{-1}$ . With the velocity gradient known, the strain rate can be calculated by using Eq. 8.15.

$$\dot{\epsilon}_i = \left( \sum_{j=1}^{NP} \frac{m_j}{\rho_{0j}} \frac{\partial W(X_i - X_j)}{\partial X_i} v_j \right) F^{-1} = \frac{\sum_{j=1}^{NP} \frac{m_j}{\rho_{0j}} \frac{\partial W(X_i - X_j)}{\partial X_i} v_j}{\sum_{j=1}^{NP} \frac{m_j}{\rho_{0j}} \frac{\partial W(X_i - X_j)}{\partial X_i} x_j}$$

In fact in the 1D case the velocity gradient is equal to the strain rate. The strain rate can now be used in an appropriate constitutive relation to obtain the Cauchy stress. As was mentioned earlier, there is no need to use a corotational stress or an objective rate since there are no material rotations in the 1D case. Because the momentum equation has been formulated in terms of the nominal stress  $P$ , the Cauchy stress needs to be transformed to the nominal stress, by using Eq. 8.17. Because in the 1D case the Jacobian of the deformation gradient  $J$  is equal to the deformation gradient, ie.  $J = F$ , it

follows that the nominal stress is equal to the Cauchy stress. Hence the only modification that is required to the momentum equation is that the kernel gradients have to be evaluated with respect to material coordinates  $X$ , and initial densities have to be used:

$$a_i = \sum_{j=1}^{NP} m_j \left( \frac{P_j}{\rho_{0j}^2} + \frac{P_i}{\rho_{0i}^2} \right) \frac{\partial W(X_j - X_i)}{\partial X_i} \quad (\text{Eq. 8.18})$$

### 8.3 Simulation Results

#### 8.3.1 Wave Propagation in Elastic Bar

In order to validate the Lagrangian kernel SPH code the 1D block wave propagation test which was also used in Chapter 3 and Chapter 4 was used. The problem setup of this test consists of two 0.5 meter long of bars that are initially in contact. A prescribed velocity of 0.001 cm/ $\mu$ s is applied for a limited time of 35 $\mu$ s to one of the free ends of the bar. This results in a stress pulse (block wave) travelling through the bar. The material model is a linear elastic model. The bar is discretised with 100 particles and the smoothing length is equal to 1.2 times the particle spacing. The resulting velocity profile in the bar after 128.8 $\mu$ s are shown in Figure 8.2, the analytical solution is also shown. The Eulerian kernel SPH results obtained in Chapter 3 are repeated for comparison. The results obtained with Lagrangian kernel show slightly less oscillation than the results obtained with a Eulerian kernel. The relatively poor shape of the square wave after it has travelled through the contact interface is due to the coarse discretisation of 100 particles that was used. The dispersion in the wave from is due to the gradual increase in contact stiffness with penetration.

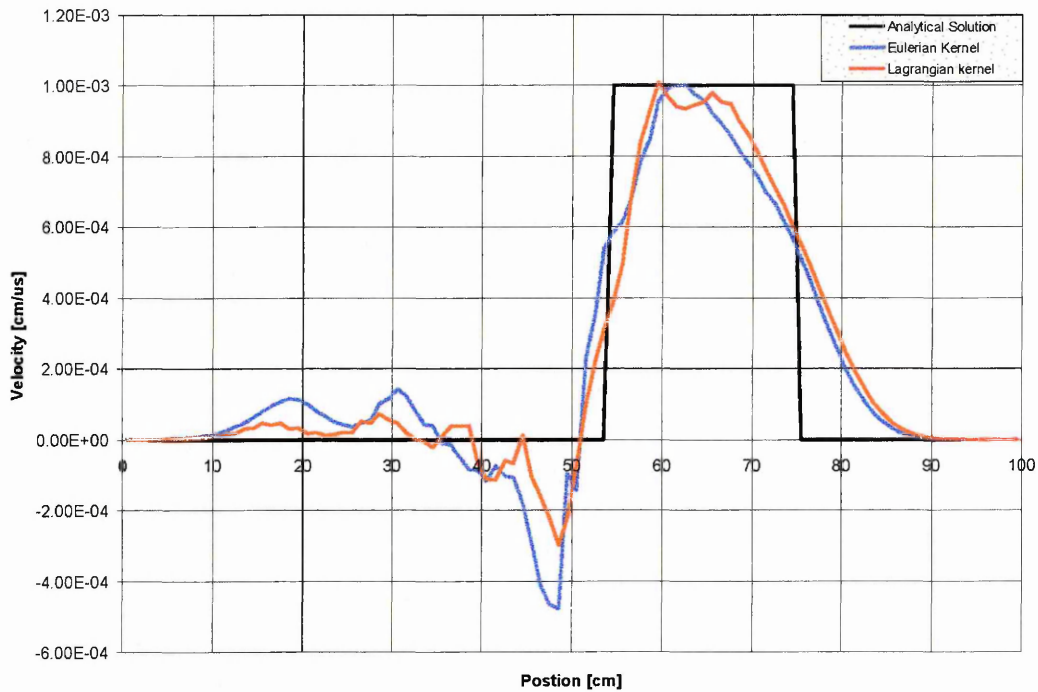


Figure 8.2 Velocity Profile for Wave Propagation Problem with Lagrangian Kernel, at time = 128.8 $\mu$ s.

### 8.3.2 Symmetric Bar Impact

The performance of the Lagrange SPH code can be further tested by simulating the 1D elastic bar impact problem used in Chapter 3. The test problem consists of a symmetrical impact of two 50.0 cm long steel bars, see Figure 3.6. 100 particles were placed along the 1D line, 50 particles in each bar. The smoothing length was 1.0 cm and the initial inter-particle space was equal to the smoothing length. The space between the two contact particles was 1.0 cm, so there was no initial penetration. Each bar was given an equal but opposite initial velocity of 200 m/s.

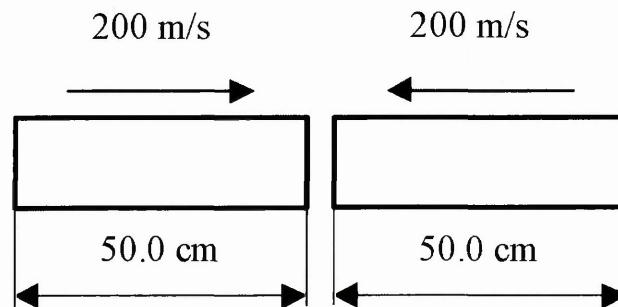


Figure 8.3: Schematic representation of the 1D test problem

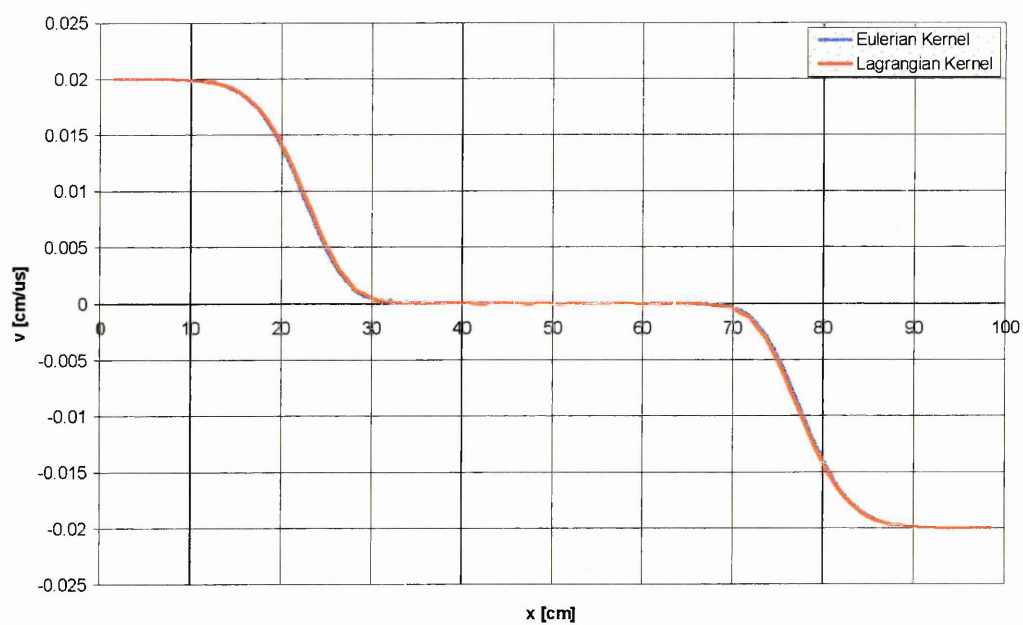


Figure 8.4: SPH velocity profiles with Lagrangian and Eulerian SPH at 48.0  $\mu\text{s}$ .

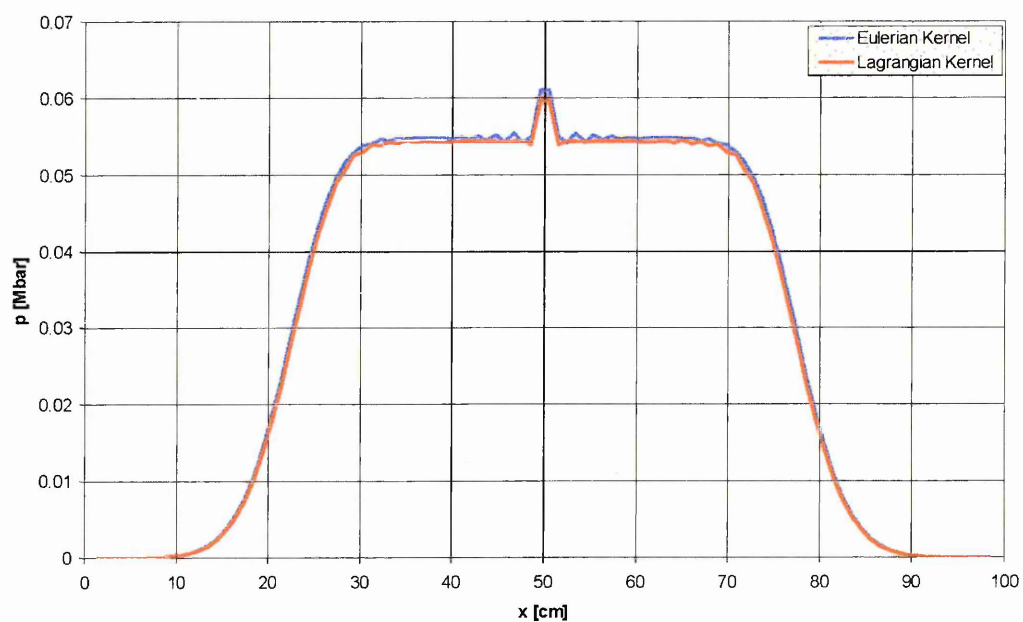


Figure 8.5: SPH pressure profiles with Lagrangian and Eulerian SPH at 48.0  $\mu\text{s}$ .



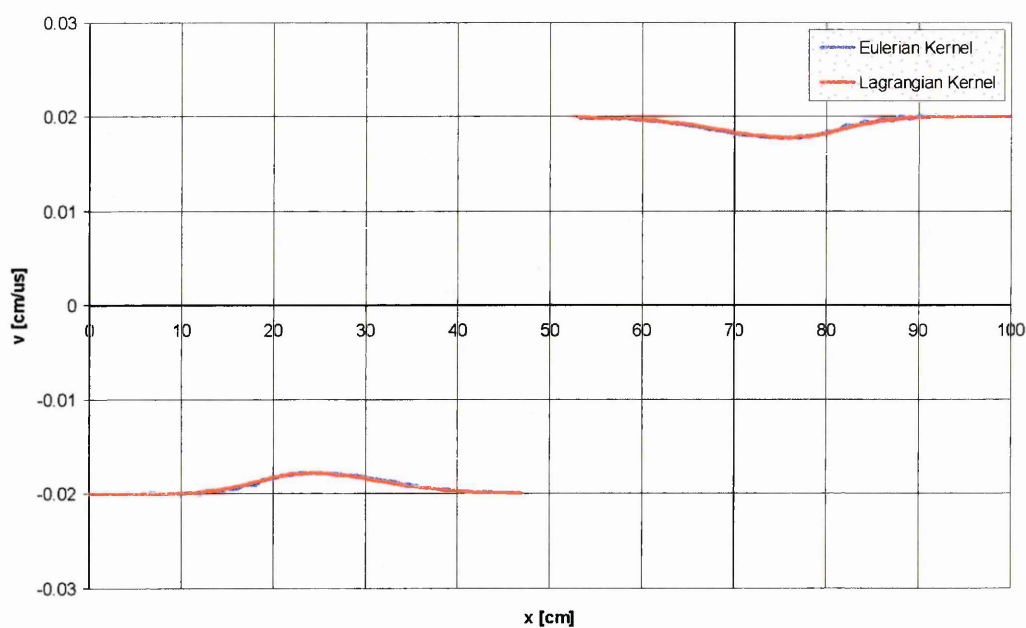


Figure 8.6: SPH velocity profiles with Lagrangian and Eulerian SPH at  $128.8\mu\text{s}$

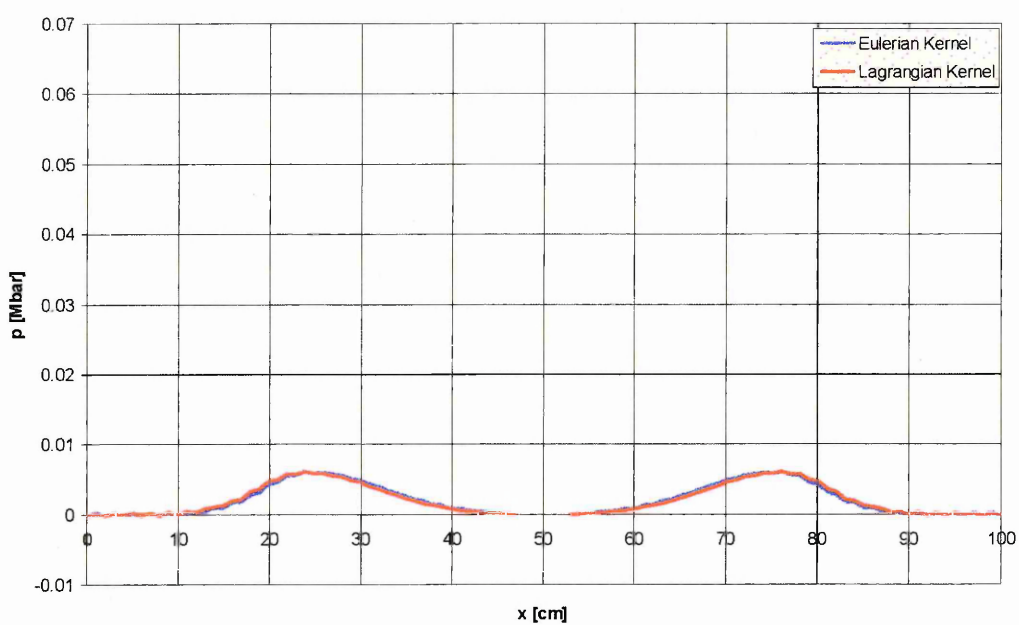


Figure 8.7: SPH pressure profiles with Lagrangian and Eulerian SPH at  $128.8\mu\text{s}$

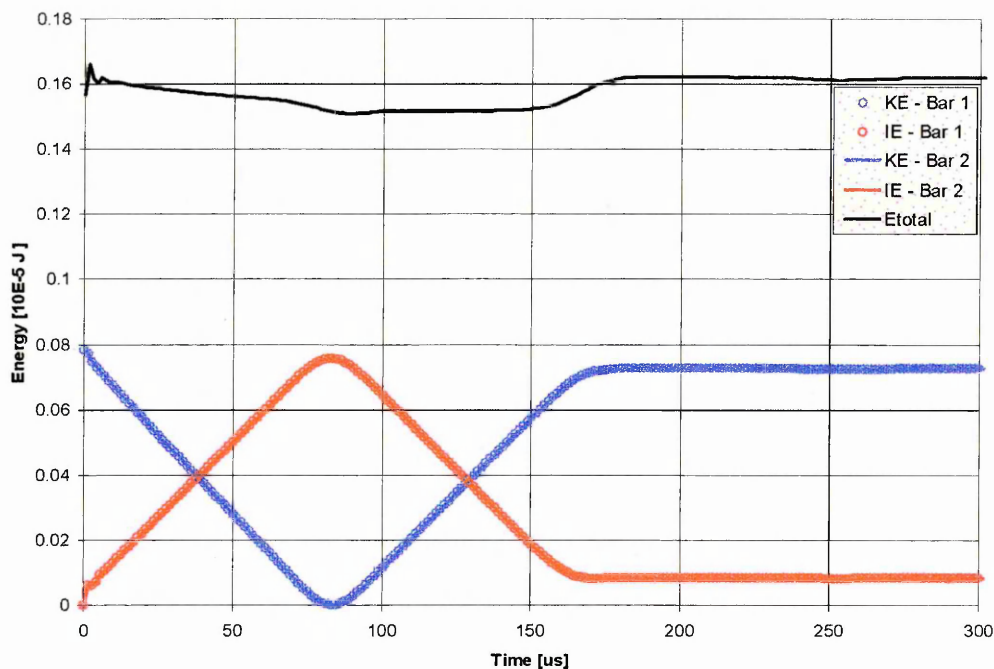


Figure 8.8: Kinetic - and Internal Energy Time Histories with Lagrangian SPH.

To provide a reference point the results obtained with Eulerian kernel simulation of Chapter 3 are repeated in Figure 8.4 to Figure 8.7. Figure 8.4 shows the resulting pressure and velocity profiles at time 48.0 $\mu$ s. The stress waves are propagating away from the point of contact. At this point no significant differences can be seen between the two sets of results. However in Figure 8.5 one can see that the oscillations in the pressure profile that are visible when using a Eulerian kernel are not present when using the Lagrangian kernel SPH code. Figure 8.6 and Figure 8.7 show the velocity and pressure profiles at time 128.8 $\mu$ s. No significant differences can be observed between the two sets of results. Finally Figure 8.8 shows the exchange between kinetic - and strain energy as a function of time for the Lagrangian kernel SPH simulation, total energy is conserved.

### 8.3.3 Plate Impact Simulation

The plate impact simulation used in Chapter 5 provides a more demanding test for the Lagrangian kernel SPH code. In order to simulate the recorded stress time histories one needs to be able to deal with tensile release waves. Since the use of a Lagrangian kernel should remove the tensile instability, one can expect better results than when using SPH with a Eulerian kernel.

The three plates were modelled with a total of 458 particles. A contact interface was defined between the Al 6082-T6 flyer plate and the Al 7010-T6 target plate. The bond between the target plate and the PMMA block was dealt with through the kernel sums.

When using a Eulerian kernel no meaningful results could be obtained due to tensile instability problems. In Figure 8.9, Figure 8.10 and Figure 8.11 the experimental and

simulation results with a Lagrangian kernel are presented. The general shape of the stress traces compares very well with the experiments. In the simulation results show a slight overestimation of the wave propagation speed. One can clearly see the different HEL's that are obtained when the material is impacted in different directions. Furthermore there is good agreement of the general pulse shape and Hugoniot stress level. All results are summarised in Table 1.

In Figure 8.9 the results for the 234 m/s impact are presented. No spall occurs at this impact velocity, consequently the characteristic pull back signals are not present. The results obtained with the SPH code are very similar to the finite element results. Both show good general agreement with the experimental results. The difference in the HEL is clearly present in both simulations. The SPH simulation predicts 0.41 and 0.37 GPa in the rolling and transverse directions respectively, compared to 0.39 and 0.33 GPa in the experiment, which is in reasonable agreement with the experiment. The Hugoniot stress levels are in very good agreement with the experiment the SPH simulations predicting a level of 0.66 compared to 0.66 GPa measured in the experiment.

In Figure 8.10 the simulated gauge traces at the higher impact velocity of  $450 \text{ m s}^{-1}$  (at two times the HEL) are presented in which clear spall signals are resolved. Again one can observe a difference in HEL, depending on the direction of impact. The Hugoniot stress levels are 1.30 GPa for the simulation and 1.3 and 1.4 GPa, depending on direction of impact, in the experiment. Again, this is in very good agreement. One can clearly see pull back signals in both traces which indicate spall. The measured values of the pull back signal (spall) in the experiment at  $450 \text{ m s}^{-1}$  are 0.31 GPa in the longitudinal orientation and 0.45 GPa in the short transverse. The values of the pull back signal in the simulation using the principal stress criterion are not really affected by the direction of impact, and are around 0.32 GPa and 0.35 GPa for rolling and transverse directions. The difference in the starting point of the spall signal is present in the simulation results, however the point does not correspond to the point at which the pullback signals start in the experimental results.

At the  $895 \text{ m s}^{-1}$  impacts the picture is similar. The general shape of the pulse agrees quite well with the experiment (see Figure 8.11), the Hugoniot stresses also compare well. In the experiment values of 3.25 GPa in the longitudinal direction and 2.8 GPa in the transverse direction are recorded. The simulation results predict a value of 2.83 GPa. For the comparison of the spall signals the situation again similar to the  $450 \text{ m s}^{-1}$  impact. In the simulation the pull back signals are 0.16 GPa (rolling) and 0.14 GPa (transverse), while the experiments record 0.37 GPa in the rolling and 0.18 GPa in the transverse.

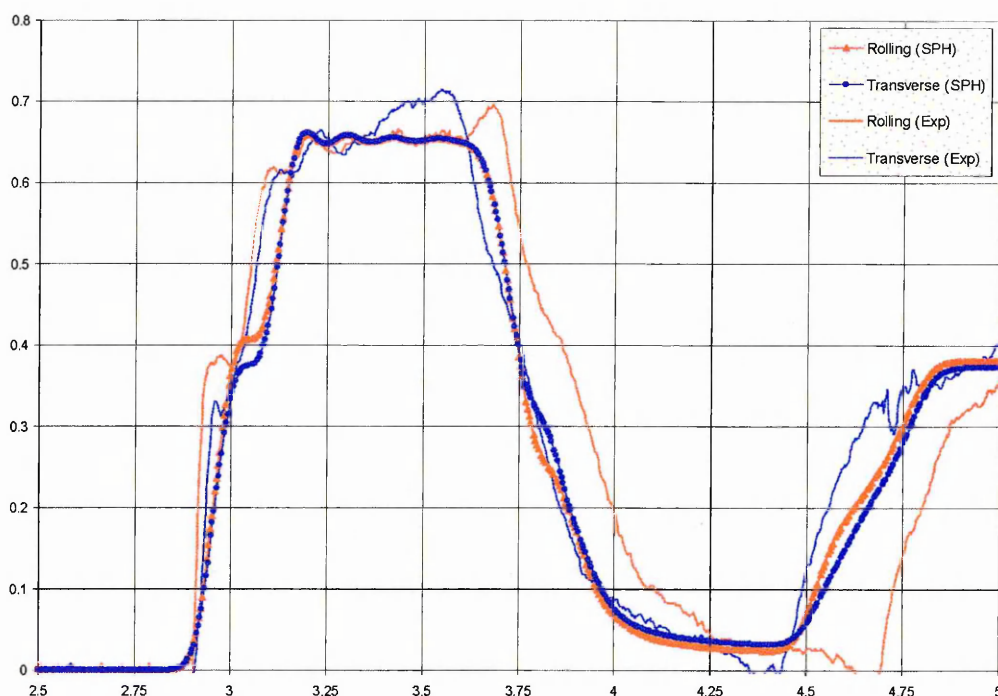


Figure 8.9: Lagrangian Kernel SPH Plate Impact Simulations - 234ms.

The main conclusion one can draw from these 1D SPH simulation results is that the SPH method has the potential of simulating this type of wave propagation and failure in anisotropic solids. If the simulation results of a 3D SPH simulation can match the results obtained of the 1D simulation then the SPH method could be a more powerful tool in testing new material models than conventional FE methods. The reason for this is that it should be easier to model failure or crack formation and propagation in SPH.

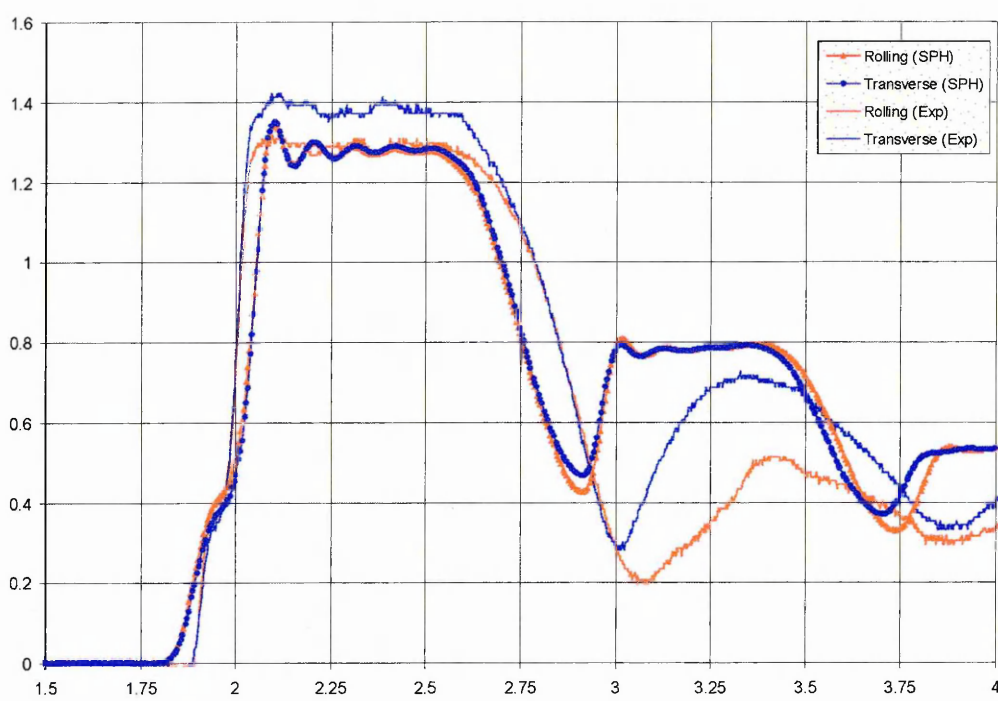


Figure 8.10: Lagrangian Kernel SPH Plate Impact Simulations - 450ms.

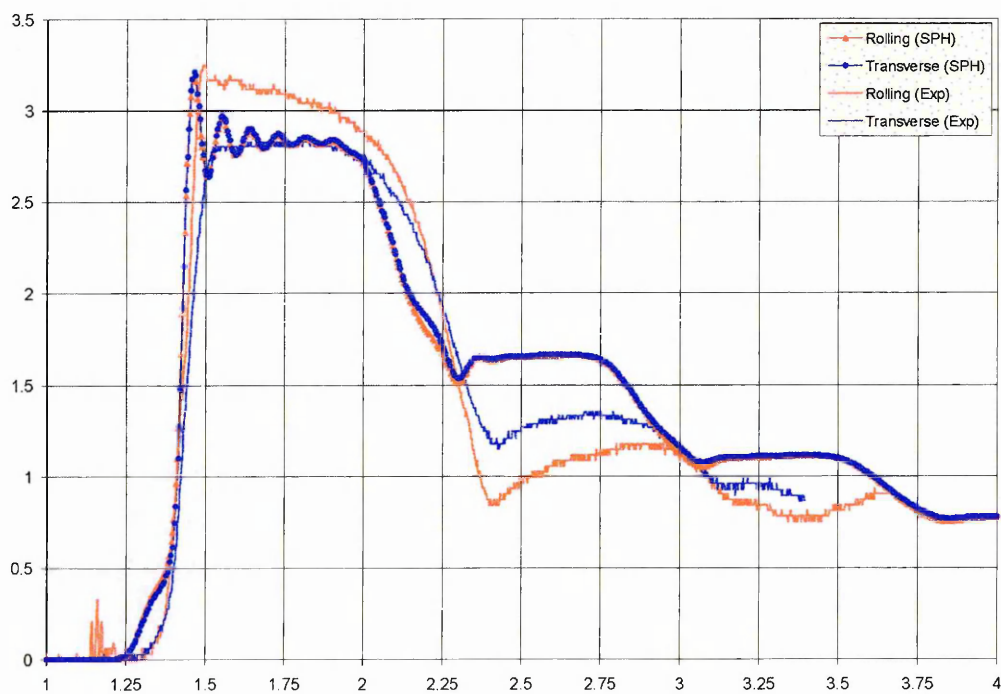


Figure 8.11: Lagrangian Kernel SPH Plate Impact Simulations - 895ms.

Vel.	Variable	Rolling			Transverse		
		Exp	FE	SPH	Exp	FE	SPH
234 [m/s]	HEL [GPa]	0.39	0.43	0.41	0.33	0.39	0.37
	Hugoniot Stress [GPa]	0.66	0.63	0.66	0.66	0.63	0.66
	Spall Strength [GPa]	N/A	N/A	N/A	N/A	N/A	N/A
450 [m/s]	HEL [GPa]	0.39	0.43	0.42	0.33	0.39	0.39
	Hugoniot Stress [GPa]	1.30	1.28	1.30	1.40	1.28	1.30
	Spall Strength [GPa]	0.31	0.20	0.35	0.45	0.20	0.32
895 [m/s]	HEL [GPa]	N/A	N/A	N/A	N/A	N/A	N/A
	Hugoniot Stress [GPa]	3.25	2.90	2.83	2.80	2.90	2.83
	Spall Strength [GPa]	0.37	0.16	0.15	0.18	0.15	0.14

Table 8.1: Summary of Results

## Chapter 9 Conclusion

### 9.1 Conclusion

The problem of hydrocode modelling of water impact has been addressed in the previous chapters. The principal work and conclusions of this thesis are:

- ▶ The treatment of contact in meshless methods has been addressed through the development of a contact algorithm which does not require the construction of surfaces. The algorithm is based on the definition of a contact potential.
- ▶ An SPH solver has been incorporated into LLNL-DYNA3D. The interaction of finite elements and SPH particles is accounted for by using a novel approach in treating the finite element nodes as particles in the contact treatment. The same contact algorithm developed for the treatment of contact in the SPH method has been used.
- ▶ A material model that takes account of anisotropy in the elastic and plastic regimes, strain-rate dependency and non-linear behaviour at high pressures including spall failure was developed.
- ▶ The FE-SPH coupling was validated against experimental data for the case of water impact of rigid cylinders on water.
- ▶ The FE-SPH coupling was further validated by demonstrating that it can be used as a tool to analyse the crash behaviour of subfloor designs on water. This was achieved by simulating the impact on water of a structure representative of an aircraft subfloor. The effect of material anisotropy, skin thickness and skin failure on the structural response was demonstrated.
- ▶ An explicit time integration 1D Lagrangian kernel SPH code was developed. It was shown that the use of a Lagrangian kernel allows the SPH method to be used for materials under tension.

### 9.2 Recommendations for further Work

Further work can improve the capability of the coupled DYNA3D-SPH code in several areas:

- ▶ The SPH contact treatment and FE-SPH coupling should be extended with an algorithm that allows tied contact to be handled accurately and robustly. A further extension of this work would be to implement an adaptive FE-SPH code which converts failed elements into SPH particles.
- ▶ The contact treatment between bodies with very different discretisation densities or shells with small thickness can be improved by implementing a pinball algorithm.

- ▶ The anisotropic material model could be improved by adding yield surfaces and hardening rules that are based on micro-mechanical behaviour.
- ▶ The anisotropic material model could be improved by adding damage models to allow for more accurate modelling of material failure.
- ▶ The accuracy of the stress update algorithm can be improved by implementing a return mapping algorithm.
- ▶ The extension of the 1D Lagrangian kernel SPH to 3D combined with an algorithm to track crack propagation, would make the simulation of dynamic brittle fracture problems possible.



## References

- [ 1 ] Anghileri M. (2001): 'Generic Water Impact Tests', D5.5.1, CAST (Crashworthiness of Helicopters on Water: Design of Structures Using Advanced Simulation Tools), EU Framework V project.
- [ 2 ] Atluri S., Zhu T.: 'A new meshless Local Petrov Galerkin (MLPG) approach in computational mechanics', *Computational Mechanics*, Vol 22, p117-127.
- [ 3 ] Atluri S., Kim H. G., Cho J. Y.: 'A critical assessment of the truly meshless local Petrov Galerkin (MLPG) and local boundary integral equation (LBIE)', *Computational Mechanics*, Vol 24, p348-372.
- [ 4 ] Attaway S., Heinstein M., Swegle J. (1994): 'Coupling of Smooth Particle Hydrodynamics with the Finite Element Method', *Nuclear Engineering and Design*, Vol. 150, p199-205.
- [ 5 ] Balsara D. (1995) : 'Von Neumann Stability Analysis of Smoothed Particle Hydrodynamics – Suggestions for Optimal Algorithms', *J. Comp. Phys.*, Vol. 121, p357-372.
- [ 6 ] Belytschko T., Lu Y. Gu L. (1994), 'Element Free Galerkin Methods', *Int. J. Num. Meth. Engrg.*, Vol 37, p229-256.
- [ 7 ] Belytschko T., Liu W.K., Moran B. (2000), 'Nonlinear Finite Elements for Continua and Structures', John Wiley & Sons, New York.
- [ 8 ] Belytschko T., Guo Y., Liu W. K., Xiao P. (2000), 'A Unified Stability Analysis of Meshless Particle Methods', *Int. J. Num. Meth. Engrg.*, Vol. 48, p1359-1400.
- [ 9 ] Belytschko T., Xiao S. (2002), 'Stability Analysis of Particle Methods with Corrected Derivatives', <http://www.tam.northwestern.edu/xiaop/stable.html>.
- [ 10 ] Belytschko T., Neal M. (1991): 'Contact-Impact By Pinball Algorithm With Penalty And Lagrangian Methods', *Int. J. Num. Meth. Engrg.*, Vol. 31, p547-572.
- [ 11 ] Belytschko T. and Yeh I. (1993), 'The splitting pinball method for contact-impact problems', *Comp. Methods Appl. Mech. Eng.* Vol. 105, p375-393.
- [ 12 ] Benson D., Hallquist J. (1990): 'A single surface contact algorithm for the post-buckling analysis of shell structures', *Comp. Methods Appl. Mech. Eng.*, Vol. 78, p141-163.
- [ 13 ] Bonet J., Lok T. (1999): 'Variational and Momentum Preservation Aspects of Smooth Particle Hydrodynamic Formulations', *Comp. Methods Appl. Mech. Eng.*, Vol. 180, p97-115.
- [ 14 ] Bonet J., Kulasegaram S. (2000): 'Correction and Stabilisation of Smooth Particle Hydrodynamic Methods with Applications in Metal Forming Simulations', *Int. J. Num. Meth. Engrg.*, Vol 47, p1189-1214.
- [ 15 ] Bonet J., Kulasegaram S. (2001): 'Remarks on tension instability of Eulerian and Lagrangian corrected smooth particle hydrodynamics (CSPH) methods', *Int. J. Num. Meth. Engrg.*, Vol 52, p1203-1220.
- [ 16 ] Bonet J., Kulasegaram S. (2002): 'Alternative Total Lagrangian Formulations for Corrected Smooth Particle Hydrodynamics (CSPH) Methods in Large Strain Dynamic Problems', preprint.
- [ 17 ] Campbell J. (1998): 'Lagrangian Hydrocode Modelling of hypervelocity Impact on Spacecraft' PhD. Thesis, Cranfield University, Cranfield, UK.]

- [ 18 ] Campbell J., Vignjevic R., De Vuyst T. (2002): 'Modelling of Hydrodynamic Ram using Smoothed Particle Hydrodynamics', in Proceedings of the 5th International Conference on Dynamics and Control of Systems and Structures in Space, Cambridge, UK.
- [ 19 ] Campbell J. , Vignjevic R., Libersky L. (2000): 'A Contact Algorithm For Smoothed Particle Hydrodynamics', *Comp. Methods Appl. Mech. Eng.*, Vol. 184, p67-85.
- [ 20 ] Campbell. P.M. (1989): 'Some new algorithms for boundary value problems in smooth particle hydrodynamics', Technical Report DNA-TR-88-286, Mission Research Corporation.
- [ 21 ] Carpenter N., Taylor R., Katona M. (1991): 'Lagrange constraints for transient finite element surface contact', *Int. J. Num. Meth. Engrg.* Vol. 32, p103-128.
- [ 22 ] Chen J., Su J., Liu. L. (2000): 'SPH Simulation of River Ice Dynamics', *J. Comp. Phys.*, Vol. 165, p752-770.
- [ 23 ] Chen J., Pan C., Wu C., Liu W.K. (1996): 'Reproducing Particle Methods for Large Deformation Analysis of Nonlinear Structures', *Comp. Meth. Appl. Mech. Eng.*, Vol. 139, p195-227.
- [ 24 ] Chen J., Wang H. (2000): 'New Boundary Condition Treatments in Meshfree Computation of Contact Problems', *Comp. Methods Appl. Mech. Eng.*, Vol. 187, p441-468.
- [ 25 ] Chen J., Beraun J., Jih C. (1999): 'An Improvement for Tensile Instability in Smoothed Particle Hydrodynamics', *Computational Mechanics* Vol. 23, p279-287.
- [ 26 ] Christon M., Voth T. (1998): 'Results of von Neumann Analyses for Reproducing Kernel Semi-Discretisations', to appear in *Int. J. Num. Meth. Engrg.*
- [ 27 ] Cleary P., Monaghan J. (1999): 'Conduction Modelling using Smoothed Particle Hydrodynamics', *Journal of Computational Physics*, Vol 148, p227-264.
- [ 28 ] Codd (2003): 'Sobolev Spaces', <http://www.maths.anu.edu.au/~codd/sobolev.pdf>.
- [ 29 ] [Cook R., Malkus D., Plesha M. (1989): "Concepts and applications of Finite Element Analysis", 3<sup>rd</sup> Edition, Wiley & Sons, USA.]
- [ 30 ] Dyka C., Ingel R. (1995): 'An Approach for Tensile Instability in Smoothed Particle Hydrodynamics', *Computers and Structures*, Vol. 57, p573-580.
- [ 31 ] Dilts G. (1999): 'Moving Least Squares Particle Hydrodynamics I: Consistency and Stability', *Int. J. Num. Meth. Engrg.*, Vol. 44, p1115-1155.
- [ 32 ] Dilts G. (2000): 'Moving Least Squares Particle Hydrodynamics II: Consistency and Stability', *Int. J. Num. Meth. Engrg.*, Vol. 48, p1503-1524.
- [ 33 ] Grady (1988): 'The Spall Strength of Condensed Matter', *J. Mech. Phys. Solids*, Vol. 36, p353-384.
- [ 34 ] Gray G.T., Bourne N.K., Zocher M.A., Maudlin P.J., Millett J.C.F. (1999), 'Influence of Crystallographic Anisotropy on the Hopkinson Fracture of Zirconium', in *Shock Compression of Condensed Matter*, Furnish M., Chhabildas L., Hixson R., Editors. 2000, AIP Press, Woodbury, NY. p. 509-512.
- [ 35 ] Gray G.T., Lopez M.F., Bourne N.K., Millett J.C.F., Vecchio K.S. (2001). 'Influence of Microstructural Anisotropy on the Spallation of 1080 Eutectoid Steel', in Thadhani, Horie Y. Eds., *Shock Compression of Condensed Matter*, AIP Press, Woodbury, NY.

- [ 36 ] Gingold R.A., Monaghan J.J. (1977): 'Smoothed particle hydrodynamics: Theory and application to non-spherical stars', *Monthly Notices of the Royal Astronomical Society*, Vol. 181, p375-389.
- [ 37 ] Gunther F., Liu W.K. (1998), 'Implementation of Boundary Conditions for Meshless Methods', *Comp. Methods Appl. Mech. Eng.*, Vol. 163, p205-230.
- [ 38 ] Gutfraind R., Savage S. (1997): 'Smoothed Particle Hydrodynamics for the Simulation of Broken-Ice Fields: Mohr-Coulomb-Type Rheology and Frictional Boundary Conditions', *J. Comp. Phys.*, Vol. 134, p203-215.
- [ 39 ] Hallquist J. (1976): 'A Procedure For The Solution Of Finite Deformation Contact-Impact Problems By Finite Element Methods', University of California, Lawrence Livermore National Laboratory, Rept. UCRL-52066.
- [ 40 ] Hallquist J. (1977): 'A Numerical Procedure For Three-Dimensional Impact Problem', American Society of Civil Engineering, Preprint 2956.
- [ 41 ] Hallquist J. (1983): 'Theoretical manual for DYNA3D'. Technical Report UCID-19401, Lawrence Livermore National Laboratory.
- [ 42 ] Hallquist J., Goudreau G., Benson D. (1985): 'Sliding Interfaces With Contact-Impact In Large-Scale Lagrangian Computation', *Comp Methods Appl. Mech. Eng.*, Vol. 50, p163-180.
- [ 43 ] Hernquist L. (1993): 'Some Cautionary Remarks about Smooth Particle Hydrodynamics', *Astrophysics J.*, Vol. 404, p717-722.
- [ 44 ] Hill R. (1950), 'The Mathematical Theory of Plasticity', Oxford University Press, Oxford.
- [ 45 ] Hughes T. (2000): 'The Finite Element Method', Dover Publications, New York, 2000.
- [ 46 ] Johnson G. (1994): 'Linking of Lagrangian particle methods to standard finite element methods for high velocity impact computations', *Nuclear Engineering and Design*, Vol. 150, p265-274.
- [ 47 ] Johnson G., Beissel S. (1996): 'Normalised smoothing functions for SPH impact computations. International Journal for Numerical Methods in Engineering, Vol. 39, p2725-2741, 1996.
- [ 48 ] Johnson G., Cook W. (1983): 'A Constitutive Model and Data for Metals Subjected to Large Strains, High Strain Rates and High Temperatures', in *Proceedings of the Seventh International Symposium on Ballistics*, The Hague, The Netherlands, p541-547.
- [ 49 ] Johnson G., Holmquist T. (1988): 'Evaluation of cylinder-impact test data for constitutive model constants', *J. Appl. Phys.*, Vol. 68, Nr. 8, p3901-3910.
- [ 50 ] Jones N. (1989), 'Some Comments on the Modelling of Material Properties for Dynamic Structural Plasticity', *Int. Con. Mech. Prop. Materials at High Rates of Strain*, Oxford, UK.
- [ 51 ] Kinslow (), *Hugoniot data handbook*.
- [ 52 ] [Krieg R., Key S. (1976), 'Implementation of a Time Independent Plasticity Theory into Structural Computer Programs', *Constitutive Equations in Viscoplasticity: Computational Engineering Aspects*, ASME, Vol. 20., p125-137.]
- [ 53 ] Kohlgrueber D. (2001): 'Survey of Software for Crash Analysis', D1.2, CAST (Crashworthiness of Helicopters on Water: Design of Structures Using Advanced Simulation Tools), D2.1, EU Framework V project., DLR, Germany.
- [ 54 ] Lancaster P., Salkauskas K., 'Surfaces Generated by Moving Least Square Methods', *Mathematics of Computation*, Vol. 37, p141-158, 1980.

- [ 55 ] Laursen T., Simo J. (1993): 'A Continuum-Based Finite Element Formulation For The Implicit Solution Of Multi-body, Large Deformation Frictional Contact Problem', *Int. J. Num. Meth. Engrg.*, Vol. 26 (1993).
- [ 56 ] Laursen T., Chawla V. (1997): 'Design Of Energy Conserving Algorithm For Frictionless Dynamic Contact Problem', *Int. J. Num. Meth. Eng.* Vol. 40, p863-886.
- [ 57 ] [Lemaitre J., Chaboche J.-L. (1990): 'Mechanics of Solid Materials', Cambridge University Press, Cambridge, England. ]
- [ 58 ] Libersky L., Petschek A. (1990): 'Smooth Particle Hydrodynamics with strength of materials', In Trease H., Fritts M., and Crowley W., editors, *Advances in the Free-Lagrange Method*, p248-257, Springer-Verlag.
- [ 59 ] Libersky L., Petschek A., Carney T., Hipp J., Allahdadi F. (1993): 'High strain Lagrangian hydrodynamics', *Journal of Computational Physics*, Vol. 109, p67-75.
- [ 60 ] Libersky L., Randles P.: 'A Spatially Staggered Particle Method Using the Strong Form', H.A. Mang, F.G. Rammerstorfer, J. Eberhardsteiner (Eds.), *Fifth World Congress on Computational Mechanics (WCCM V)*, Vienna, Austria, 2002
- [ 61 ] Lin J. (1999): *Dyna3D Users' Manual*, Lawrence Livermore National Laboratory.
- [ 62 ] Liu W.K., Chang H. Chen J.S., Belytschko T. (1988), 'Arbitrary Lagrangian and Eulerian Petrov-Galerkin Finite Elements for Nonlinear Problems', *Computer Methods in Applied Mechanics and Engineering*, Vol. 68, p259-310.
- [ 63 ] Liu W.K., Chen Y. (1995): 'Wavelet and Multiple Scale Reproducing Kernel Method', *International Journal of Numerical Methods in Fluids*, Vol. 21, p901-933.
- [ 64 ] Liu W.K., Chen Y., Uras R., Chang C. (1996): 'Generalised Multiple Scale Reproducing Kernel Particle Methods', *Computer Methods in Applied Mechanics and Engineering*, Vol. 139, p91-158.
- [ 65 ] Liu W.K., Hao S., Belytschko T., Li S., Chang C. (1999): 'Multiple Scale Meshfree Methods for Damage Fracture and Localisation', *Computational Materials Science*, Vol 16, p197-205.
- [ 66 ] Liu W.K., Jun S., Li S., Adey J., and Belytschko T.. (1995): 'Reproducing kernel particle methods for structural dynamics', *International Journal for Numerical Methods in Engineering*, Vol. 38, p1655-1679,
- [ 67 ] Liu W.K., Jun S., Li S., Adey J., and Belytschko T. (1997): 'Reproducing kernel particle methods for nearly incompressible hyperelastic solids' Submitted to *International Journal for Numerical Methods in Engineering*.
- [ 68 ] Liszka T.J., Duarte C.A.M., Tworzydło W.W. (1996), 'hp Meshless Cloud Method', *Comp. Meth. Appl. Mech. Engrg.*, Vol. 139, p263-288.
- [ 69 ] Lucy L. (1977): 'A numerical approach to the testing of the fission hypothesis', *The Astronomical Journal*, Vol. 82, no. 12, p1013-1024, 1977.
- [ 70 ] Monaghan J. (1982): 'Why particle methods work' *SIAM Journal on Scientific and Statistical Computing*, Vol. 3, no. 4, p422-433.
- [ 71 ] Monaghan J. (2000): 'SPH without a Tensile Instability', *Journal of Computational Physics*, Vol. 159, p 290-311.
- [ 72 ] Moshe E., Eliezer S., Dekel E., Ludmirsky A., Henis Z., Werdiger M., Eliaz N., Eliezer D. (1998): 'An increase in the spall strength in aluminium, copper and metglass at strain rates larger than  $10^7 \text{ s}^{-1}$ ', *J. Appl. Phys.*, Vol. 83, p4004-4011.
- [ 73 ] Needleman A. (1988): 'Material Rate Dependent and Mesh Sensitivity in Localisation Problems', *Computer Methods in Applied Mechanics and Engineering*, Vol. 67, p68-85.

- [ 74 ] Needleman A. (1989): 'Dynamic Shear Band Development in Plain Strain', *Journal of Applied Mechanics*, Vol. 56, p1-9.
- [ 75 ] Papapostolou D. (2002), 'Bi-Axial Testing on Metallic Specimen', D5.1, CAST (Crashworthiness of Helicopters on Water: Design of Structures Using Advanced Simulation Tools), EU Framework V project., NTUA, Athens.
- [ 76 ] Randles P., Petschek A., Libersky L., Dyka C. (2001): 'Stability of DPD and SPH', in *Lecture Notes in Computational Science and Engineering*, Proceedings of the International Workshop on Meshfree Methods for Partial Differential Equations, Bonn, Germany.
- [ 77 ] Randles P. and Libersky L. (1996): 'Smoothed particle hydrodynamics: Some recent improvements and applications', *Computer methods in applied mechanics and engineering*, Vol. 139, p375-408.
- [ 78 ] Rosenberg Z., Luttwak G., Yeshurun Y., and Partom Y. (1983): 'Spall studies of differently treated 2024A1 specimen', *J. Appl. Phys.* Vol. 54, p2147-2152.
- [ 79 ] Rosenberg Z., Yaziv D., and Partom Y. (1980): 'Calibration of foil-like manganin gauges in planar shock wave experiments', *J. Appl. Phys.*, Vol. 51, p3702-3705.
- [ 80 ] Sauer M. (2000), 'Adaptive Kopplung des Netzf freien SPH-Verfahrens mit Finiten Elementen zur Berechnung von Impaktvorgängen', PhD Thesis, Universität der Bundeswehr München.
- [ 81 ] Simo J., Tarnow N. (1992): *The Discrete Energy-Momentum Method Conserving Algorithm for Non-linear Elastodynamics*, *Z. Angew. Math. Phys.*, Vol. 43, Birkhauser Verlag, Basel.
- [ 82 ] Stryk R., Johnson G. and Beissel. S. (1996): 'SPH for high velocity impact computations' *Computer methods in applied mechanics and engineering*, Vol. 139, p347-373.
- [ 83 ] Swegle J., Attaway S., Heinsteins M., Mello F., and Hicks D. (1994): 'An analysis of smoothed particle hydrodynamics', Technical Report SAND93-2513, Sandia National Laboratories.
- [ 84 ] Smallman R.E. (1985): *Modern Physical Metallurgy*, 4th ed. Butterworths, London.
- [ 85 ] Takeda H., Miyama S., Sekiya M. (1994): 'Numerical simulation of viscous flow by smooth particle hydrodynamics', *Progress of Theoretical Physics*, Vol. 92, p939-960.
- [ 86 ] Tedoldi P. (2001), 'Report on Accident Survey', D1.1, CAST (Crashworthiness of Helicopters on Water: Design of Structures Using Advanced Simulation Tools), EU Framework V project, Agusta, Italy.
- [ 87 ] Vignjevic R., Campbell J., Libersky L. (2000): 'A Treatment of Zero-Energy Modes in the Smoothed Particle Hydrodynamics Method', *Computer Methods in Applied Mechanics and Engineering*, Vol. 187, p67-85.
- [ 88 ] Wood handbook (1999) *Wood as an engineering material*, Gen. Tech. Rep. FPL-GTR-113, Forest Products Laboratory, Madison, WI U.S. Department of Agriculture, Forest Service, Forest Products Laboratory, Sawyer County, Wisconsin.
- [ 89 ] Zerilli F., Armstrong R. (1987): 'Dislocation-mechanics-based Constitutive Relations for Material Dynamics Calculations', *J. App. Phys.*, Vol. 61, Nr. 5, p1816-1825.
- [ 90 ] Zheng J., Wang Z.(1995): 'Spall Damage in Aluminium Alloy', *Int. J. Solids Structures*, Vol. 32, p1135-1144.

- [ 91 ]    Zhong Z. (1993): 'Finite Element Procedures For Contact-Impact Problem', Oxford Science Publications, Oxford.
- [ 92 ]    Zhu T., Atluri S. (1998): 'A Modified Collocation Method and a Penalty Formulation for Enforcing the Essential Boundary Conditions in the Element Free Galerkin Method', Computational Mechanics, Vol. 21, p211-222.
- [ 93 ]    Zienkiewics O. (1991): 'The Finite Element Method', McGraw Hill, London.
- [ 94 ]    Zukas J. (1990), 'High Velocity Impact Dynamics', Wiley and Sons, New York.

## APPENDIX A. Classical SPH derivation

The classical SPH derivation of the conservation of momentum equation usually starts from:

$$a = \frac{1}{\rho} \nabla \cdot \sigma = \nabla \cdot \frac{\sigma}{\rho} + \frac{\sigma}{\rho^2} \cdot \nabla \rho \quad \text{in } \Omega \quad (\text{Eq. A.1})$$

and one obtains:

$$\langle a_i \rangle = - \int_{\Omega} \nabla W \cdot \left( \frac{\sigma}{\rho} \right) dV - \frac{\sigma_i}{\rho_i^2} \cdot \int_{\Omega} \nabla W \rho dV. \quad (\text{Eq. A.2})$$

Because the derivation starts with the density term in the LHS of the equation, and the application of the chain rule, there is an extra term,

$$- \frac{\sigma_i}{\rho_i^2} \cdot \int_{\Omega} \nabla (\delta w) \rho dV$$

and also one obtains the nodal smoothed value  $\langle a_i \rangle$  for the acceleration immediately, also there is a density factor in the first integral on the LHS.

Instead it will be shown that in the Galerkin formulation one has a weighted integral in the LHS,

$$\int_{\Omega} \delta w \rho a dV$$

which will later result in the mass matrix.

One could also start the ‘classical’ SPH derivation from:

$$\rho a = \nabla \cdot \sigma \quad \text{in } \Omega \quad (\text{Eq. A.3})$$

which would result in

$$\int_{\Omega} W \rho a dV = - \int_{\Omega} \nabla W \cdot \sigma dV \quad (\text{Eq. A.4})$$

and by linearising the RHS of the equation:

$$\langle \rho_i a_i \rangle = - \int_{\Omega} \nabla W \cdot \sigma dV \quad (\text{Eq. A.5})$$

The approximation of the density by

$$\langle \rho_i \rangle = \sum_j m_j W(x_i - x_j) \quad (\text{Eq. A.6})$$

is commonly mentioned in the literature [ 13 ], [ 14 ], [ 70 ]. However this equation does not perform very well [Hernquist].

The LHS of the Eq. A.5 be approximated using the standard SPH interpolation of a gradient:

$$\langle \rho_i a_i \rangle = - \sum_j \frac{m_j}{\rho_j} \nabla_j W(x_i - x_j) \cdot \sigma_j \quad (\text{Eq. A.7})$$

$$\langle \rho_i a_i \rangle = \sum_j \frac{m_j}{\rho_j} \nabla_i W(x_i - x_j) \cdot \sigma_j \quad (\text{Eq. A.8})$$

or

$$\langle a_i \rangle = \sum_j \frac{1}{\rho_i} \frac{m_j}{\rho_j} \nabla_i W(x_i - x_j) \cdot \sigma_j \quad (\text{Eq. A.9})$$

This is different from the equations that are usually used to calculate the accelerations. The two most commonly used expressions are:

$$\langle a_i \rangle = \sum_j m_j \frac{\sigma_j \pm \sigma_i}{\rho_i \rho_j} \cdot \nabla_i W(x_i - x_j) \quad (\text{Eq. A.10})$$

$$\langle a_i \rangle = \sum_j m_j \left( \frac{\sigma_j}{\rho_j^2} + \frac{\sigma_i}{\rho_i^2} \right) \cdot \nabla_i W(x_i - x_j) \quad (\text{Eq. A.11})$$

The acceleration calculation in Eq. A.11 is variationally consistent with Eq. A.6 for the density, while Eq. A.10 is variationally consistent with [ 13 ]:

$$\dot{\rho} = -\rho \operatorname{div}(v).$$

The first of these expressions is obtained by using the fact that:

$$\int \nabla W dV = 0 \quad (\text{Eq. A.12})$$

in the discretised form this expression becomes:

$$\sum_j \frac{m_j}{\rho_j} \nabla W(x_i - x_j) = 0 \quad (\text{Eq. A.13})$$



In this discretised form this expression does not in general hold when using a SPH kernel. However this condition can be enforced by applying kernel normalisation [ 14 ], [ 77 ] or a reproducing kernel correction [ 66 ].

The second expression is obtained by starting from:

$$a = \frac{1}{\rho} \nabla \cdot \sigma = \nabla \cdot \frac{\sigma}{\rho} + \frac{\sigma}{\rho^2} \cdot \nabla \rho \quad \text{in } \Omega \quad (\text{Eq. A.14})$$

The stress difference formulation has a stabilising influence on the results as it preserves momentum exactly.

In the context of an explicit time-integration of the conservation equations, to simulate impact and wave propagation problems, the acceleration of particle  $i$  in ‘classical’ SPH is calculated using the above equations [ 17 ], or alternatively one could use:

$$\langle a_i \rangle = \sum_j m_j \frac{\sigma_j}{\rho_i \rho_j} \frac{\partial w(x_i - x_j)}{\partial x_i} \quad (\text{Eq. A.15})$$

The stress is calculated from the constitutive equation using the current strain (in 1-D):

$$\sigma_j = E \varepsilon_j \quad (\text{Eq. A.16})$$

$$\sigma_j = E \sum_l \frac{m_l}{\rho_l} (d_l - d_j) \frac{\partial w(x_j - x_l)}{\partial x_j} \quad (\text{Eq. A.17})$$

or by leaving out the term in  $d_j$ , which was added to restore symmetry and hence conserve momentum:

$$\sigma_j = E \sum_l \frac{m_l}{\rho_l} d_l \frac{\partial w(x_j - x_l)}{\partial x_j}. \quad (\text{Eq. A.18})$$

Substituting this in Eq. A.15 results in the following expression for  $a_i$ :

$$\langle a_i \rangle = \sum_j \sum_l \frac{1}{\rho_i} \frac{m_j}{\rho_j} \frac{m_l}{\rho_l} \frac{\partial w(x_j - x_l)}{\partial x_j} \frac{\partial w(x_i - x_j)}{\partial x_i} E d_l \quad (\text{Eq. A.19})$$

# APPENDIX B. Implementation of DYNA3D-SPH Coupling

## Description of the Enight CASE Data Format

The next paragraphs will discuss the Enight data format, and describe in detail how the output section of DYNA3D was modified to write its data in Enight format.

```
FORMAT
type: ensight

GEOMETRY
model: example.geo

VARIABLES
vector per node: 1 1 displacement example_dis.ens
vector per node: 1 1 velocityexample_vel.ens
scalar per element: 1 1 efpsexample_eps.ens
tensor symm per element: 1 1 stressexample_sig.ens

TIME
time set: 1
number of steps: 50
filename start number: 1
filename increment: 1
time values: 0.00000E+00
              0.10000E+01
              0.20000E+01
              .
              .
              .
              0.50000E+02

FILE
file set: 1
filename index: 1
number of steps: 50
```

Figure B.1: Example of .case file.

There are two types of input formats for Enight. The first older format, Enight 5, result in a large number of output files to be created. Apart form a file with geometry and a file with timestep data, a new file for every variable for every plot time is also created. While this format does allow for vector data to be stored as one output variable, it does not allow tensor data to be stored in one file. Hence, for a typical analysis with, for example, 10 output variables (displacement, velocity, 6 stress components, eff. plastic strain, and density) and 50 plots would result in 500 output files to be created. The second Enight format, CASE, overcomes these problems and therefore this format has been implemented.

The CASE format consists of a set of data files and one file, the .case file which contains references to all other data files that are part of this 'case'. An example of a .case file is shown in Figure B.1.

The case file has a fixed format and information is split into five blocks FORMAT, GEOMETRY, VARIABLES, TIME, FILE.

The first block, FORMAT, specifies the input format that the data to be visualised is written in, in this case Enight.

The second block, GEOMETRY, specifies the file name that contains the geometry data. This file contains nodal positions, element connectivity, etc.

The third block, VARIABLES, defines the variables that will be visualised. This definition consists of one definition line for every variable. First of all a statement is made about the nature of the field variable. Field variables can either be scalar, vector or symmetric 2<sup>nd</sup> order tensors. A scalar variable obviously has only one component per data point, while a vector has 3. For a symmetric 2<sup>nd</sup> order tensor 6 values are stored. The reason that a vector variable has 3 components is that Enight data always assumes to be dealing with 3D data. So if one wants to visualise 2D data a 3<sup>rd</sup> component can be added by simply completing the data with zeroes where required. The second section on the variable declaration are the reference to the time and file set that the variables belong to. In Figure B.1 this is the two numbers '1 1' that follow the variable declaration. The first number refers to the time set the variable belongs to, the second number defines the file set. The next piece of information is a description of the variable name. In the example in Figure B.1 'displacement', 'velocity', 'efps', 'stress'. The last entry is the actual filename where this variable's data is stored. There are no requirements on the name of the file, or on its file extension. In fact no file extension is required. However if the data of a variable is stored in more than one file then \*'s need to be added to indicate that there are multiple files. The data files then need to contain numbers where the \*'s are positioned. for example is the file name was 'example\_dis.\*\*\*' then the files would be expected to contain a file extension of three numbers.

The next block of data is the TIME block. This block defines the number of time sets in the case. Every time set contains a time set number, the number of time steps in that time set, the filename start number and the increment, and finally a list with all the problem times at which the data was sampled. The possibility of defining several time blocks is useful for example when the output consists of both, stateplots and time history plots. Generally the time history data would be sampled at a higher frequency. The filename start number and increment refer to the first file number. For example if the filename start number is 000 (this would assume '\*\*\*' was added in the variable definition) and filename increment is 1, then in the example the output files for the displacement would be 'example\_dis.000', 'example\_dis.001', ...

The last block of information in the .case file is the FILE block. This block is used in the case where more than one timestep is stored per file. Filename index refers to how many output file per variable there are. Hence the \*\*\* in the variable definition will in

this case be replaced by the filename index. The number of steps specifies how many time steps are stored in each file. If, as in the example in Figure B.1, the filename index is 1 then all plots for one variable are stored in one file. The number of plots that can be stored in one file is only limited by the maximum file size allowed on the specific computer. This depends on the operating system and hardware. In the case of a PC the maximum file size is 2Gb, which should be sufficient for most simulations.

The second main file in the case output format is the .geo file (Figure B.2). As was mentioned before this file contains nodal positions and element connectivity data. If multiple plots are stored in one file then these blocks of data have to be enclosed by 'BEGIN TIME STEP' and 'END TIME STEP'. If the geometry does not change the geometry only has to be written once. The first two lines in the geometry file are title lines. The following lines specify whether node and element numbers are included in the data or whether it is left up to Enight to number them internally. For the case under consideration it is essential that node and element numbers are identical to the node and element numbers in the analysis, hence the lines 'node id given', 'element id given' are written. Next the nodal coordinates are written. This section start with the line 'coordinates', on the second line the number of nodes, followed by the list of node numbers and coordinates, one per line. When a combination of FE and SPH is used, the nodes are all the element nodal coordinates followed by the particle positions.

The next section in the .geo file is the part information. Element data can be grouped in several parts, which when read into Enight can be switched on or off using a toggle switch. For the visualisation of combined FE and SPH data it was decided to assign one part to every material number. In other words every material can contain either a combination of solid, shell and beam elements, or particles only. The data format for every part is illustrated in Figure B.2. The first part line is simply 'part' followed by the part number. The second line is a description line, which is followed by the blocks of element data, grouped per element type. For example (see Figure B.2), the lines 'hexa8', number of hexa8 (eight noded hexahedral elements) elements, and the list of all hexa8 element numbers and connectivity that belong to the material number that is being considered. For SPH particles the element numbers are simply the same as the SPH node numbers.

Finally the structure of a data file will be described (see Figure B.3 and Figure B.4). One has to make a distinction between a data file containing plot data of a nodal variable or an element variable. Examples of nodal variables are displacement and velocity, examples of element data are stress and effective plastic strain.

The data format for a nodal variable is shown in Figure B.3, with the displacement field as example. Again the data of individual plots is separated by 'BEGIN TIME STEP' and 'END TIME STEP'. The first line for every plot is the variable name, this name will be used as a label in the Enight GUI. Next all the nodal data is simply written out in one big block. However Enight requires that all data lines contain six values, apart from the last line which depending on the number of nodes may contain from one up to six values. It is assumed that the order in which the nodal data is written corresponds to the order of the nodes as they were specified in the .geo file. Because Enight requires every line to have six numbers on it the way data is written out depends on the field

variable. If the field is scalar then nodal data of six nodes will be written on one line, whereas if the field is a vector field data of only two nodes will be written on every line. In the case on a symmetric second order tensor (six unique components) one data line will contain the data of only one node.

```

BEGIN TIME STEP
Title1
Title2
node id given
element id given
coordinates
  625
    1 0.00000E+00 0.00000E+00 0.00000E+00
    2 0.00000E+00 0.00000E+00 0.10000E+00
    .
    .
    .

  625 0.10000E+00 0.10000E+00 0.10000E+00
part
material
hexa8
  100
    1 1 66 79 14 2 67 80 15
    2 66 131 144 79 67 132 145 80
    .
    .
    .
    .
    .

  100 246 311 324 259 247 312 325 260
quad4
  100
    1 11 166 179 114
    2 166 131 244 179
    .
    .
    .
    .
    .

  100 304 211 224 159
part 2
SPH particles in part 2
point
  300
  326 326
  .
  .
  .

  625 625
END TIME STEP

```

Figure B.2: Example of .geo file.

The output format for element variables is shown in Figure B.4, with the effective plastic strain as example. Again the data of individual plots is separated by 'BEGIN

TIME STEP' and 'END TIME STEP'. The first line for every plot is the variable name, this name will be used as a label in the Ensight GUI. The difference with nodal data is that in the case of element data the data is written out grouped by part (ie. material number in this case), and for every part the data again needs to be written out grouped by element type. This way the element data has to be written out in the same order as the element definitions in the .geo file.

```
BEGIN TIME STEP
displacement
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
. . . ..
. . . ..
. . . ..

0.00000E+00 0.00000E+00 0.00000E+00
END TIME STEP
.
.
.

BEGIN TIME STEP
displacement
0.10000E+00 0.10000E+00 0.10000E+00-0.10000E+00-0.10000E+00-0.10000E+00
0.20000E+00 0.20000E+00 0.20000E+00-0.20000E+00-0.20000E+00-0.20000E+00
. . . ..
. . . ..
. . . ..

0.30000E+00 0.30000E+00 0.30000E+00
END TIME STEP
```

Figure B.3: Example of nodal variable data file.

```

BEGIN TIME STEP
efps
part 1
hexa8
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
. . . . .
. . . . .
. . . . .

0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
part 2
point
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
. . . . .
.. . . .
. . . . .

0.00000E+00 0.00000E+00
END TIME STEP
.
.
.

BEGIN TIME STEP
efps
part 1
hexa8
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
. . . . .
. . . . .
. . . . .

0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
part 2
point
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
.. . . .
. . . . .
. . . . .

0.00000E+00 0.00000E+00
END TIME STEP

```

Figure B.4: Example of element variable data file.

### ***Description of the DYNA3D Data Structure***

The aim of this paragraph is to give an overview of where and in which format the data that needs to be written to the output files is stored. From the description of the data structure in section 1.2.2 it is clear that all data will either be accessed via common

blocks, pointers to the main database a(\*), or data can be calculated locally in a subroutine. The global analysis information is stored in common /bk00/. Some of the data stored in this common is the number of nodes numnp, the number of solid elements numelh, the number of beam elements numelb, the number of shell elements numels, and the number of thick shell elements numelt.

The nodal field variables and element connectivity are stored in the main database and can be accessed via the use of pointers. Table B.1 gives an overview which pointer corresponds to what data. These pointers are stored in a common block, COMMON /bk13/. The nodal positions are stored in ascending order in the main database, this is illustrated in Table B.2.

Data	Pointer
Nodal Position	lc11
Nodal Velocity	lc9
Nodal Acceleration/Force	lc10
Nodal Mass/Inverse Mass	lc14
Solid Element Connectivity	lc1h
Beam Element Connectivity	lc1b
Shell Element Connectivity	lc1s
Thick Shell Element Connectivity	lc1t

Table B.1: Overview of Pointers to Nodal Field Variables and Element Connectivity Data in the Main Database.

a(lc11)	a(lc11+1)	a(lc11+2)	a(lc11+3)	a(lc11+4)	a(lc11+5)	...
x(1,1)	x(2,1)	x(3,1)	x(1,2)	x(2,2)	x(3,2)	...

Table B.2: Storage of Nodal Positions in the Main Database.

The element connectivity is stored in a less convenient format for output. The output has as the first entry the material number of the element, followed by the nodal connectivity. This sequence is then repeated for all elements of that type, ie. solid, beam, shell or thick shell. There are however different 3 different types of solid and two types of shell elements. A solid element can be either a hexahedral (8 nodes, brick), a pentahedral (6 nodes, wedge) or tetrahedral (4 nodes, pyramid). These elements are illustrated in Figure B.5. A shell element can be either a quadrilateral (4 nodes) or triangular element (3 nodes) (see Figure B.6).

It is important to be aware that the order in which the element connectivity data is stored in the main database does not correspond to the order in which the elements are defined in the input file. This means that for example the element with element number one will not necessarily carry element number one in the database. In other words every element has an input element number and an internal element number. The reason for this is unclear. In order to be able to link input and internal element numbers DYNA3D has an array of pointers that point from the input element number to the internal element number. These arrays of pointers are defined for every element type. They are called nhxpnt, nshpnt, nbmpnt, and ntspt for solid, shell, beam and thick shell elements respectively. As an example if nhxpnt(2) is 307, then this means that the element which has been defined as a hexahedral element with element number 2, has the internal



element number 307. These arrays with pointers are stored in DYNA3D's main database. In order to access these arrays with pointers DYNA3D defines another 4 pointers nh13, nb13, ns13 and nt13 that point to the starting position of the nhxpnt, nshpnt, nbmpnt, and ntspt arrays in the main database a(\*). These pointers nh13, nb13, ns13 and nt13 are stored in a common block, common /sorter/.

The solid element connectivity is stored in a sequence of 9 numbers per element. The first number is the material number of the element. This is followed by eight numbers for the nodal connectivity. In the case of a hexahedral element the sequence of the nodes is as illustrated in Figure B.5 and Table B.3. For pentahedral and tetrahedral elements DYNA3D still allocates and writes eight node numbers but some of the nodes are repeated in the sequence of nodes. From Figure B.5 and Table B.4 it can be seen that for pentahedral elements the 5<sup>th</sup> node is written to the 6<sup>th</sup> and 9<sup>th</sup> position, and the 6<sup>th</sup> node is written to the 7<sup>th</sup> and 8<sup>th</sup> position. For tetrahedral elements (see Figure B.5 and Table B.5) the 3<sup>rd</sup> node is written to the 4<sup>th</sup> and 5<sup>th</sup> positions, and the 4<sup>th</sup> node is written to the remaining positions, ie. from the 6<sup>th</sup> to the 9<sup>th</sup> position.

a(lc1h)	a(lc1h+1)	a(lc1h+2)	a(lc1h+3)	a(lc1h+4)	a(lc1h+5)	a(lc1h+6)	a(lc1h+7)	a(lc1h+8)	...
mat(n)	node1(n)	node2(n)	node3(n)	node4(n)	node5(n)	node6(n)	node7(n)	node8(n)	...

Table B.3: Storage of Hexahedral Element Connectivity in the Main Database

a(lc1h)	a(lc1h+1)	a(lc1h+2)	a(lc1h+3)	a(lc1h+4)	a(lc1h+5)	a(lc1h+6)	a(lc1h+7)	a(lc1h+8)	...
mat(n)	node1(n)	node2(n)	node3(n)	node4(n)	node5(n)	node6(n)	node6(n)	node5(n)	...

Table B.4: Storage of Pentahedral Element Connectivity in the Main Database.

a(lc1h)	a(lc1h+1)	a(lc1h+2)	a(lc1h+3)	a(lc1h+4)	a(lc1h+5)	a(lc1h+6)	a(lc1h+7)	a(lc1h+8)	...
mat(n)	node1(n)	node2(n)	node3(n)	node3(n)	node4(n)	node4(n)	node4(n)	node4(n)	...

Table B.5: Storage of Tetrahedral Element Connectivity in the Main Database.

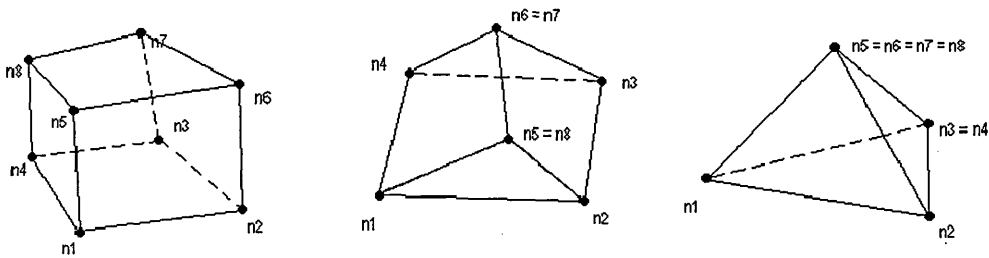


Figure B.5: Types of Solid Elements.

In the case of shell elements 5 positions are reserved for the element connectivity per element in the main database. Again the first position of the connectivity data refers to the material number of the element. The remaining four positions are the nodal connectivity (Table B.6). In the case of a triangular shells the 3<sup>rd</sup> node is written to positions 4 and five (Table B.7).

a(lc1s)	a(lc1s+1)	a(lc1s+2)	a(lc1s+3)	a(lc1s+4)	...
mat(n)	node1(n)	node2(n)	node3(n)	node4(n)	...

Table B.6: Storage of Quadrilateral Shell Element Connectivity in the Main Database.

a(lc1s)	a(lc1s+1)	a(lc1s+2)	a(lc1s+3)	a(lc1s+4)	...
mat(n)	node1(n)	node2(n)	node3(n)	node3(n)	...

Table B.7: Storage of Triangular Shell Element Connectivity in the Main Database.

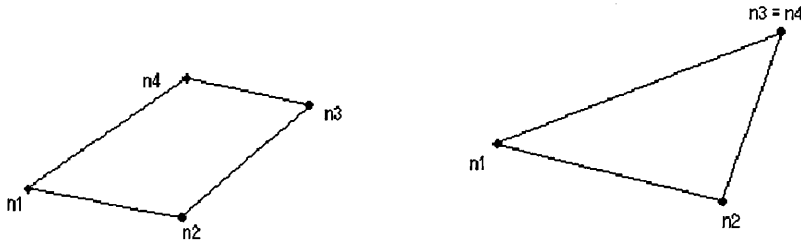


Figure B.6: Types of Shell Elements.

For beam elements 6 positions are reserved for the element connectivity data of each element in the main database. The first number is again the material number of the element. This is immediately followed by the nodal connectivity of the beam element in positions 2 and 3. The fourth entry is the node which is specified to define the orientation of the cross section of the beam.

a(lc1s)	a(lc1s+1)	a(lc1s+2)	a(lc1s+3)	a(lc1s+4)	a(lc1s+5)	...
mat(n)	node1(n)	node2(n)	node3(n)	-	-	...

Table B.8: Storage of Beam Element Connectivity in the Main Database.

As can be seen by comparison with the LLNL-DYNA user manual [ 61 ], these formats differ from the input format. DYNA3D reorganises the nodal connectivity input format before it is written to the main database. This is done in the routines elemnh, elemns, elemnb, elemnt which are the routines read the connectivity in from the input file. With this reorganisation of the nodal connectivity it is easy to determine which material an element belongs to, and which type of element it is (hexahedral, tetrahedral, ...). The type can be determined by checking if nodes in the nodal connectivity are repeated. For example, if one is looking a solid element and the 3<sup>rd</sup> and 4<sup>th</sup> node in the connectivity data for that element are identical then the element is a tetrahedral element.

Finally, element data such as stresses and effective plastic strains are usually the main variables that need to be visualised during post-processing of analysis results. Therefore the stresses and effective plastic strains need to be extracted from the main database.

DYNA3D uses pointers to access the element stresses stored in the main database. Different pointers are used for every element type. These pointers are lc15, ns06, nb06 and nt04 for solid, shell, beam and thick shell elements respectively. When a(lc15), a(ns06), a(nb06) or a(nt04) is an argument in a call to a subroutine then the corresponding

dummy variable is usually called auxvec. This means that the auxvec array will contain the stresses of one type of element. So the individual element stresses have to be extracted from this array auxvec. In order to do this yet another set of pointers is defined in DYNA3D. Again there are different pointers for every element type. These pointers are lochvh, lochvs, lochvb and lochvt for solid, shell, beam and thick shell elements respectively. The reason these pointers are required is because the stresses are not simply stored in the same order as the element input numbers, nor are they stored in the same sequence as the element connectivity data is written. The pointers lochvh, lochvs, lochvb and lochvt do not however point at the location of the stress data of an input element number, rather it points from the element's internal element number to the location of that element's stress. This can be best clarified with an example. One can take solid elements as an example, but the concept extends equally to the other element types. If one wants to find out where the stress of an element that was defined as, say, element 9 in the input file is stored, then one has to use two sets of pointers to arrive at the right location in the auxvec array. First of all the nhxpnt pointer will connect the input and internal element numbers, for input element number 9, nhxpnt(9) = 147. Then one uses the lochvh pointer to find the position of the stress in auxvec, in the example: lochvh(147) = 12438. This means that the stress of element with input element number 9 will be stored from auxvec(12438) onwards. This is illustrated schematically in Figure B.7. Finally the lochvh, lochvs, lochvb and lochvt pointers are stored in the main database. The pointers to the locations of these respective arrays are nh14, ns14, nb14 and nt14.

Other element variables can be stored in a similar way. For example the effective plastic strain is stored after the last stress component in auxvec. So in the example the stress components of element 9 would be stored in auxvec(12438), ..., auxvec(12443), and the effective plastic strain of this element in auxvec(12444).

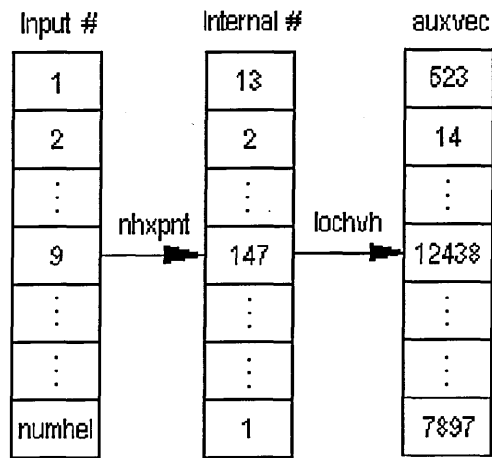


Figure B.7: Pointers from input element number to position stress data in auxvec.

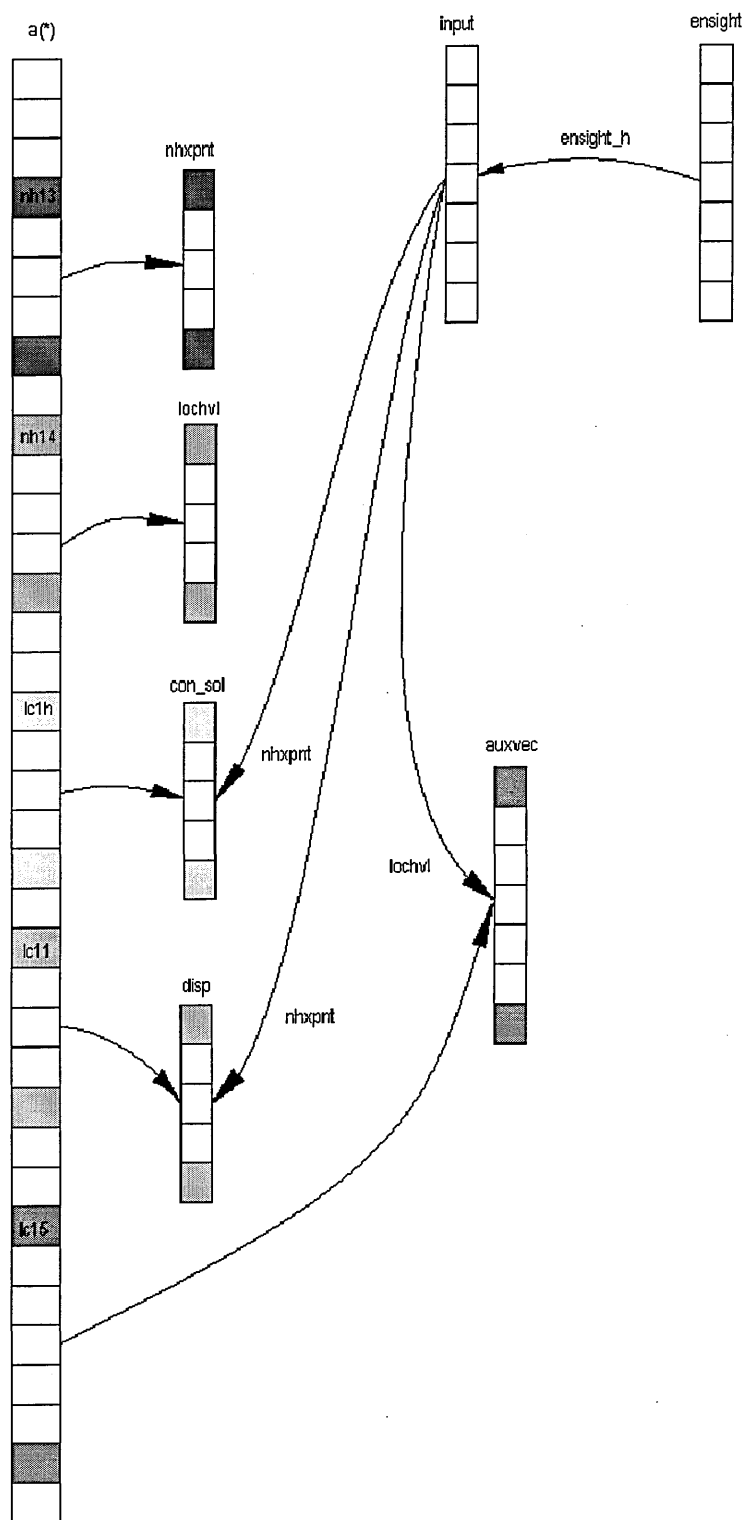


Figure B.8: DYNA3D Data Structure.

A last important variable that has to be accessed when writing output is the activity flag. this flag is equal to 1 when the element is active, but is set to 0 when the material is

removed from the calculation. This happens for example when a material with failure option is used. When the material fails this flag is set to 0 and the stresses are set to 0. The element is then removed from the remainder of the calculation. This element activity flag is stored in the database a(\*), and similar to the nodal connectivity, is grouped per element type. The names of the pointers of the element activity flag are listed in Table B.9.

Element Type	Pointer to Element Activity Flag
Solid	nh17+numelh
Shell	ns17+numels
Beam	nb17+numelb
Thick Shell	nt17+numelt

Table B.9: Pointer to Element Activity Flag

As a summary a schematic representation of how data is stored in the main database, and how it is linked to input data is shown in Figure B.8.

**Description of the MCM Data Structure**

All particle data in MCM is available via MODULES. Inside the modules scalar, vector and tensor particle variables are stored in the same format irrespective of the variable. The format for these three physical variable types is shown in Table B.10, np stands for the number of particles.

Particle Variable Type	Array Format
Scalar	array(1:np)
Vector	array(1:3, 1:np)
Tensor	array(1:3, 1:3, 1:np)

Table B.10: MCM data format for Particle Variables.

For example the density is stored in rho(1:np), displacement is stored in disp(1:3,1:np), the stress is stored in sigma(1:3, 1:3, 1:np).

**Description of the Combined DYNA3D-SPH Output Section**

From the previous two sections it was made clear how the data required for combined output of LLNL-DYNA3D and SPH analyses can be accessed in the DYNA and SPH databases. Using this information, the output section in the main solution loop in DYNA3D, in subroutine fem3d, was modified such that the DYNA3D information and the SPH data is written out in Ensight format. The state output section in DYNA3D is called in fem3d with a call to subroutine prtdat. This routine creates the output for the dynamic relaxation and transient solution sections. This routine was replaced with several calls to routines that write the case write\_case, geometry write\_geo, and output variable files write\_ensight\_3d\_init, ensightdat, write\_ensight\_3d\_close. The variables that are written out are the nodal displacements and velocities, and the element stresses and effective plastic strain. Because neither DYNA3D nor MCM stores the nodal displacements, the nodal coordinates are updated every timestep, it was necessary to add an extra routine that stores the initial positions for both the FE nodes and SPH particles.

This routine, called `mcm_write_x0`, was added in the section of the `fem3d` routine that is used only for the first timestep.

The routines that write the case and geometry files are called every time a stateplot is written. This is necessary because the case file contains the number of plots, and the problem times of each plot. The geometry file needs to be written every time in order to be able to deal with changing geometry, particularly element deletion. By checking the value of the element activity flag, only the element connectivity of the elements that have not been deleted are written to the geometry file. The case file follows the same format as described in Figure B.1.

The first sections in the `write_geo` file are straightforward, the required title lines and node and element id lines and the total number of nodes (FE nodes + SPH particles), are written. This is followed by a loop that writes out the positions of the FE nodes and SPH particles. During this loop the number of SPH nodes per material is also calculated.

Because of the differences in the nodal connectivity output format this connectivity data has to be reordered or relabelled for the output. The element connectivity data in the output format is sorted per material number and then per element type, with input element numbering. The element connectivity data structure in DYNA3D is that this data is grouped per element type, and with an internal element numbering, while in MCM the particles are stored per input particle number (element number and particle number are the same). The nodal connectivity output section in the `write_geo` routine loops over the material number, every material is written out as a new part. In order to write out all data in the correct order a loop over all elements for all the element types (solid, shell, beam and thick shell) first counts the number of elements of every type for the current material. This is necessary because at the start of each element block in a particular part Enight requires the number of elements of that type to be given. This loop has to be performed every time a geometry file is written to allow for the deletion of elements. With the number of each element type known for the current material the subroutine loops over all elements once more, this time to write out the element connectivity. While writing out the element connectivity another array of pointers is written. This array of pointers has the purpose of storing the order in which the element connectivity is written out, and they form a link between the position in the `.geo` file and the input element number (see Figure B.8). This is useful to reduce the effort required to write out element field variable data, as these are written out in the same order and grouping (material number and element type) as the connectivity data, and without element number. These arrays of pointers have the following general form:

$$\text{ensight\_}*(\text{num},j) = k.$$

In this the `*` can be either 'h', 's', 'b' or 't' depending on whether solid, shell, beam or thick shell elements are being written, so there are 4 arrays in total. The `k` is the input element number, `j` is the material, and `num` is the position of the element in the `ensight` data file. The `ensight_*` files are dimensioned as:

$$\text{ensight\_}*(1:\text{numel}, 1:\text{nummat}).$$

The final part of the geometry file writes the particle element numbers, which are the same as the particle numbers.

The change of nodal and particle positions alone would not require a geometry file to be written every time plot data was created. The movement of the geometry could be visualised because the nodal and particle displacements are written as an output variable. However in order to be able to deal with element deletion or remeshing it is necessary to rewrite the element connectivity data every time a plot dump is made. Because the applications that are being considered in this thesis may involve element deletion (due to material failure or excessive distortion of the element) or remeshing, the geometry file is written for every plot dump.

The next routine which is called in the state dump section is `write_ensight_3d_init`. This routine has as only purpose to create or to open and append the nodal and element field variable output files.

The main routine in the output of field variables is the `ensightdat` routine. This routine is a copy of the original output routine `prtdat`, but contains the necessary modifications to output the data in Ensight format rather than to flush data into the `d3plot` files.

The nodal displacements, velocities and accelerations are written out in a new routine `write_ensight_disp`, which is called at from `ensightdat`. The nodal positions, velocities and accelerations and the number of FE nodes are passed as arguments into this routine. The routine first calculates the nodal displacements, and then writes the displacements and velocities to their respective output files. This routine only writes the FE data, the SPH data will be added later via another subroutine.

In order to output the stress another routine `write_ensight_stress` was added. The original `prtdat` routine contains four calls to routines that write the stress to the `d3plot` files. The routines are called `prtr`, `tshprt`, `shprt`, `prtbem` for solid, thick shell, shell and beam stresses respectively. The `write_ensight_stress` routine contains calls to four modified versions of the `prtr`, `tshprt`, `shprt`, `prtbem` routines which are called `prtr_ensight`, `tshprt_ensight`, `shprt_ensight`, `prtbem_ensight`. The purpose of adding the `write_ensight_stress` routine in between is to extract to make sure that when the `prtr_ensight` (or similar) routine is called it is given the right element data to extract out of the database. Therefore the input element number of the element that needs to be written to the Ensight output file is extracted from the `ensight_*` array and passed to the `prtr_ensight` (or similar) routine. The `write_ensight_stress` routine also keeps track of the position of the data on the current line in the output file (Ensight requires 6 values on every data line). This number is also passed to the `prtr_ensight` routine (or similar).

The `prtr_ensight`, `tshprt_ensight`, `shprt_ensight`, `prtbem_ensight` can not simply be copies of `prtr`, `tshprt`, `shprt`, `prtbem`, but need to be modified as well. The reason for this is that the original routines write out blocks of data. The blocks have size `lnv`, which is defined as `42*lnv`. Instead the loop which writes the data for this block of elements is removed, and only the element number that is passed in the call statement is written. This means that these routines are now called for every element that is written.

The stress is stored in the `auxvec`, which is a dummy variable in the `prtr`, `tshprt`, `shprt`, `prtbem` routines. By using the `n*pnt` and `lochr*` arrays, the pointer `lav` is assigned the correct value to point at the starting position in `auxvec(*)` of the element data that has to be written. The six stress components are stored from `auxvec(lav)` to `auxvec(lav+5)`. Via a call to subroutine `blkcpy` these numbers are copied to a temporary array `sig`, which is then used in the write statement to write the data to the output file. Since there are six values, output to Enight is easy as every element uses one complete line.

The effective plastic strain is also stored in `auxvec`, in the position after the six stress components. In the routines we are considering the effective plastic strain is copied to `sig(7)`, this done in subroutine `blkcpy` at the same time as the stress is copied to `sig`. The effective plastic strain is then written to its Enight output file. As mentioned earlier, the position of this number on the current line in the output file is a dummy variable in the `prtr_ensight`, `tshprt_ensight`, `shprt_ensight`, `prtbem_ensight` routines. If it is the sixth number on one line an end-of-line character is written and the position variable is reset, otherwise the end-of-line character is omitted.

The output for shell elements is almost identical. The shell stresses and effective plastic strains are written by a call to subroutine `shprt_ensight`, from `ensightdat`. This routine, `shprt_ensight`, only unpacks data by a call to `sprt_ensight`. The only difference is that for shell elements there are several stresses to consider. DYNA3D allows for the output of the midplane, and upper and lower fibre stresses of the element, if the trapezoidal integration rule is used. Alternatively DYNA3D can also output the element stresses at the bottom-most, middle and upper-most integration point through the thickness of the element. These stresses are all stored in a local array `stress(*)`. The `sprt_ensight` is a modified version of the original `sprt` subroutine such that only the stress of one specified element is written to the Enight stress output data file.

### ***Coupling Algorithm***

In the previous sections it was described how the solution loops of an FE solver and SPH solver can be combined, and how the FE and SPH results can be visualised by using an output format compatible with Enight.

This section concentrates on the actual interaction between FE elements and SPH particles. With the modelling of water impact in mind, the contact algorithm developed in Chapter 3 was used to model the interaction of elements and particles.

In order to use the particle to particle contact algorithm developed in Chapter 3 the FE domain has to be treated as particles. The most straightforward way of doing this is to treat the element nodes as particles within the contact algorithm. This is novel to compared to most published coupling algorithms. In these algorithms one usually tries to preserve the FE part and add special transition element or to add the FE integrals to the SPH interpolation. In the coupling algorithm described in this section neither FE element or SPH kernel interpolations require any modification. The interaction is handled through the calculation of a contact force vector for the elements interacting with particles, and vice versa. By modelling the interaction in this way, using a



node-to-node contact algorithm, no element data is required. This is a major advantage as no element integrals need to be evaluated, or shape functions used.

Before the contact force between a specific FE node and the SPH particle in its proximity can be determined it is first necessary to perform a neighbour search including the FE nodes. The nodal positions, stored in the DYNA3D database, are passed to all the relevant neighbour search routines. The linked list search requires the minimum and maximum coordinate values of any SPH particle or FE node, for every coordinate axis. This value is calculated in `mcm_move` (and in `mcm_initvbls` during the initialisation phase), the routine that updates the SPH particle positions. An extra loop was added there that includes the FE nodes in the determination of these minimum and maximum values. With these values known the neighbour search can be performed over particles and FE nodes.

Because the FE nodes are considered as contact particles only (no kernel sums will include the FE nodes as neighbours) it is most convenient to create a separate linked list and neighbour list for the FE nodes. These variables are called `mcm_llpointer_dyna` (the array containing the pointers to the first node in the linked list for every cell), `mcm_llgrid_dyna` (the linked list array), `mcm_contlist_dyna` (the list of FE contact node for every SPH particle). The declaration of these variables was added in the module `mcm_database`. The allocation of the memory for these variables is performed in `mcm_allocate_memory` for `mcm_llpointer_dyna`, `mcm_setuplist` for `mcm_llgrid_dyna`, and in `mcm_init_neighb` (which is called during the initialisation phase) and `mcm_neighbours` for `mcm_contlist_dyna`.

The routines that calculate the FE contact nodelist are called from `mcm_neighbours` (during the solution phase) and `mcm_init_neighb` (during the initialisation phase). The linked list is calculated in subroutine `mcm_setuplist`, and the neighbour list and contact lists are calculated in `mcm_ll_neighbours` (co-located SPH), `mcm_spvp_neighbours` (non-co-located SPH).

In order to allow for the use of the contact when symmetry planes are used it is necessary to create a symmetry contact neighbourlist for the DYNA3D nodes. The symmetry conditions on the FE nodes are of course imposed in the DYNA3D section, but these nodes are required to ensure the correct behaviour of the SPH nodes on the symmetry plane. In order to achieve this three extra variables are declared in the database `mcm_dyna_nsym`, `mcm_sym_dyna` and `par(i)%n_symdyna`. The symmetry neighbour section is included in the `mcm_init_neighb` and `mcm_neighbours` routines with calls to `mcm_set_sym_neighb` and `mcm_checksym_dyna`.

The actual contact routine `mcm_rep_cont_couple` that calculates the forces on FE nodes and SPH particle due to their interaction is added in the section in `fem3d` that deals with the contact - impact algorithm and processes the slidelines. The routine is essentially a copy of the SPH particle to particle contact routine with the difference that the contact particles are the DYNA3D nodes. The loop is over `par(i)%ncont_dyna + par(i)%n_symdyna`. The routine that calculates the *j* particle information, `mcm_get_j_dyna_info`, is a modification of the equivalent SPH subroutine `mcm_get_j_cont_info`, and is passed the nodal positions `a(lc11)` and nodal mass `xms(*)`.

The main difference between the SPH particle to particle algorithm and the FE node to particle algorithm is that in the particle to particle algorithm the loop is over all particles that are in contact, and the contact force vector is calculated for each of these nodes in turn. In the FE node to SPH particle routine the loop is only over the SPH nodes. This is because no list is stored of the contact neighbours of the FE nodes. The list `mcm_contlist_dyna` is an array with for every SPH particle a list of the FE nodes that are in contact with it. Therefore not only has the contribution of FE node  $j$  to the contact force on SPH particle  $i$  be calculated, but at the same time one also has to add the contribution of particle  $i$  to the contact force of FE node  $j$ . These contact forces are stored in `par(i)%repulsion_dyna` (contact force on particle  $i$ ) and `mcm_coupling_acc` (array of contact forces on FE nodes). In fact the numbers stored in these arrays are the contact forces divided by the particle or nodal mass, ie. an acceleration:

$$\begin{aligned}
 f_c^{FE}(x_i^{FE}) &= \sum_j^{NSPH} \frac{m_j^{SPH}}{\rho_j^{SPH}} \frac{m_i^{FE}}{\rho_i^{FE}} Kn \left( \frac{W(x_j^{SPH} - x_i^{FE})}{W(\Delta p_{avg})} \right)^{n-1} \nabla_{x_j^{SPH}} W(x_j^{SPH} - x_i^{FE}) \\
 a_c^{FE}(x_i^{FE}) &= \sum_j^{NSPH} \frac{m_j^{SPH}}{\rho_j^{SPH} \rho_i^{FE}} Kn \left( \frac{W(x_i^{FE} - x_j^{SPH})}{W(\Delta p_{avg})} \right)^{n-1} \nabla_{x_i^{FE}} W(x_i^{FE} - x_j^{SPH}) \quad (Eq. B.1) \\
 \Delta a_c^{FE}(x_i^{FE}) &= \frac{m_j^{SPH}}{\rho_j^{SPH} \rho_i^{FE}} Kn \left( \frac{W(x_i^{FE} - x_j^{SPH})}{W(\Delta p_{avg})} \right)^{n-1} \nabla_{x_i^{FE}} W(x_i^{FE} - x_j^{SPH})
 \end{aligned}$$

These contributions to the particle and nodal accelerations can now be added to the particle and nodal acceleration vectors. For the particles this is done in `mcm_acceleration`, the routine that calculates the accelerations due to internal forces. For the FE nodes this is done in a subroutine `mcm_coupled_acc` which is added in the main time integration loop in `fem3d` just after the accelerations due to the internal forces are calculated (labels 20 and 25). This concludes the description of the changes required to couple DYNA3D with an SPH solver in order to model fluid-structure interaction problems.

This algorithm is able to deal with the interaction of particles with solid and shell elements. The only drawback to this implementation is that the difference in resolution of FE elements and particle volume should be similar. If there would be a large difference it is possible that particles that contact the element face in the middle (far away from the nodes) do not see a large enough contact force to slow them down, and may penetrate the elements. Also if the element size is much larger than the particle size the particles will interact with the elements at a distance away from the element face.

One way to overcome this problem is to implement a the contact algorithm similar to the splitting pinball algorithm [ 11 ]. However in order to calculate the pinballs the element face nodal connectivity has to be known. This would also require the resulting pinball forces to be interpolated, via shape function on the element face, to the FE nodes.

## APPENDIX C. Detailed Description of f3dm33

The first part of the routine is concerned with the reading in and initialisation of all the material properties, and the rotation of the stress and strain rate tensors from global to material coordinates. The next step is to calculate the strains from the strain-rates by multiplying with the timestep  $dt1$ .

The material parameters  $F$ ,  $G$ ,  $H$  are calculated from the anisotropy parameters  $R$ ,  $P$ ,  $Q_{ab}$ ,  $Q_{bc}$ ,  $Q_{ca}$  using equations 4 to 9.

The next step is the initialisation of the  $A$  matrix (Eq. 5.21), which is stored in a variable dpl premultiplied by a scalar factor:

$$dpl = \begin{bmatrix} G+H & -H & -G & 0 & 0 & 0 \\ -H & F+H & -F & 0 & 0 & 0 \\ -G & -F & F+G & 0 & 0 & 0 \\ 0 & 0 & 0 & 2N & 0 & 0 \\ 0 & 0 & 0 & 0 & 2L & 0 \\ 0 & 0 & 0 & 0 & 0 & 2M \end{bmatrix} \quad (\text{Eq. C.1})$$

Using this matrix one can write:

$$\frac{\partial f}{\partial \sigma} = \sqrt{\frac{3}{2}} \frac{dpl \sigma}{\sigma(F+G+H)} \quad (\text{Eq. C.2})$$

The following line then updates the yield stress using a linear strain hardening law

$$\sigma_y = \sigma_0 + E_p^a \epsilon_{eff}^p \quad (\text{Eq. C.3})$$

The current yield stress obtained in this way is stored in the variable  $ak$ ,  $ep$  is the equivalent plastic strain that was calculated during the previous timestep at the end of the f3dm33 routine.

The final step in the initialisation section of the f3dm33 routine consists of passing the stresses and strains to new variables. The stresses are passed on from  $so1, \dots, so6$  to  $sigo1, \dots, sigo6$ , the  $o$  stands for old, from the previous timestep. The strains are passed on from  $dd1, \dots, dd6$  to  $dt1, \dots, dt6$ . Also the effective plastic strain increment vector  $d\epsilon^p$  initialised to zero, stored in variable  $dp$ . Using the stress tensor components from the previous timestep the old equivalent stress is calculated. The value of  $ak$  is then set to the bigger of the yield stress calculated using the equivalent plastic strain (Eq. 14) and the old equivalent stress  $s1old$ . This value is then passed onto  $x1$  to be used further in the routine. The hydrostatic stress is stored in  $smold$ . The increment in effective plastic strain  $d\bar{\epsilon}^p$  is initialised to zero (variable name  $depix$ ). This finishes the initial stage of this subroutine in which all the necessary variables are read in or initialised.

The next section in the program consists of the calculation of the stress and plastic strain increments together with the calculation of the history variables that have to be known in the next timestep. The procedure that is used is as follows: First of all an elastic trial stress is calculated as follows (similar to a radial return algorithm):

$$\sigma_{ij}^{\text{trial}} = \sigma_{ij}^{t_n} + C_e \Delta \varepsilon \quad (\text{Eq. C.4})$$

The equivalent trial stress is then calculated and the elastic fraction of the stress increment is calculated. If the new equivalent stress, based on the old constitutive matrix  $c_e$ , is less than the current yield stress  $\sigma_Y$  then this means that the increment was completely elastic and hence that the constitutive matrix that was used was correct. In this case the program moves straight to the latter part of the routine in which the new strains, equivalent plastic strain etc are calculated. If the equivalent stress is larger than the current yield stress however then a correction is necessary because the stress increment was not completely inside the yield surface. Corrections to the constitutive matrix will be necessary. The first step in this process is to determine which part of the stress increment was elastic. The procedure is similar to the one described in [ 29 ] in the section on the 'Tangent-Stiffness Method'. To this end two variables are used  $f_{ela}$  and  $f_{pla}$ . The elastic part of the trial stress is calculated as a fraction of the trial stress increment as follows:

$$m = \frac{\sigma_Y - \bar{\sigma}^{t_n}}{\bar{\sigma}^{\text{trial}} - \bar{\sigma}^{t_n}} \quad (\text{Eq. C.5})$$

using this ratio  $m$  the stress is updated with the elastic part:

$$\sigma_{ij}^{t_{n+1}, \text{TEMPORARY}} = \sigma_{ij}^{t_n} + m C_e \Delta \varepsilon \quad (\text{Eq. C.6})$$

If the value obtained for  $m$  does not lie between 0 and 1 then  $m$  is set 0 or 1 respectively. The plastic part fraction of the stress increment is simply  $1-m$ . This temporary stress vector is stored in  $\text{sigc}$ .

The following step is the calculation of the plastic stress increment with the correct constitutive matrix. The calculation of the constitutive matrix is performed in the subroutine  $\text{cepx}$ . The resulting elastoplastic stress-strain matrix is passed back to  $\text{f3dm33}$  as  $\text{cep}$ . The next step is to calculate the plastic part of the stress increment. In order to do this the elastoplastic constitutive matrix is calculated. The expression is, assuming associative plasticity:

$$C_{ep} = C_e - C_e \frac{\partial f}{\partial \sigma} \{C_\lambda\}^t \quad (\text{Eq. C.7})$$

where  $\{C_\lambda\}^t$  is defined by the relation of the plastic multiplier  $d\lambda$  to the strain increment:

$$d\lambda = \{C_\lambda\}^t d\epsilon \quad (\text{Eq. C.8})$$

Comparing with the general expression for  $d\lambda$ :

$$d\lambda = \frac{\frac{\partial f}{\partial \sigma_{ij}} : (C : d\epsilon_{ij})}{\frac{\partial f}{\partial \sigma_{ij}} : (C : \frac{\partial f}{\partial \sigma_{ij}}) + \sqrt{\frac{2}{3}} C_i} \quad (\text{Eq. C.9})$$

it is clear that  $\{C_\lambda\}^t$  can be expressed by:

$$\{C_\lambda\}^t = \frac{\frac{\partial f}{\partial \sigma_{ij}} : C}{\frac{\partial f}{\partial \sigma_{ij}} : (C : \frac{\partial f}{\partial \sigma_{ij}}) + \sqrt{\frac{2}{3}} C_i} \quad (\text{Eq. C.10})$$

one obtains the following expression for the elastoplastic constitutive matrix  $C_{ep}$ ,

$$C_{ep} = C - C : \frac{\frac{\partial f}{\partial \sigma_{ij}} : C}{\frac{\partial f}{\partial \sigma_{ij}} : (C : \frac{\partial f}{\partial \sigma_{ij}}) + \sqrt{\frac{2}{3}} C_i} \quad (\text{Eq. C.11})$$

In these expressions the kinematic hardening term has been left out since model 33 does not have this capability.

Now that  $C_{ep}$  is known one can calculate the new stress using the following expression:

$$\sigma_{ij}^{t_{n+1}} = \sigma_{ij}^{t_{n+1}, \text{TEMPORARY}} + (1-m) \left( C_e - C_e \frac{\partial f}{\partial \sigma_{ij}} \{C_\lambda\}^t \right) \Delta \epsilon_{ij} \quad (\text{Eq. C.12})$$

and is stored as sigt.

The next step is the calculation of the plastic strains and the equivalent plastic strain. The increment in plastic strain is calculated using Eq. C.13 [ 44 ]. eb1, eb2, eb3, eb4, eb5, eb6, dedef and dede are temporary variables.

$$d\bar{\epsilon}^{pl} = \sqrt{\left[ F \left( \frac{Gd\epsilon_y^{pl} - Hd\epsilon_z^{pl}}{FG + GH + HF} \right)^2 + G \left( \frac{Fd\epsilon_x^{pl} - Hd\epsilon_z^{pl}}{FG + GH + HF} \right)^2 + H \left( \frac{Fd\epsilon_x^{pl} - Gd\epsilon_z^{pl}}{FG + GH + HF} \right)^2 + \frac{d\epsilon_{yz}^{pl2}}{2L} + \frac{d\epsilon_{zx}^{pl2}}{2M} + \frac{d\epsilon_{xy}^{pl2}}{2N} \right]} \quad (\text{Eq. C.13})$$

The increment in plastic strain is stored in depix. The manual uses combinations of  $d\epsilon_a^p$  and  $d\epsilon_c^p$ , whereas in the code it is a combination of  $d\epsilon_a^p$  and  $d\epsilon_b^p$ . After some manipulation and the assumption:

$$d\epsilon_a^p + d\epsilon_b^p + d\epsilon_c^p = 0 \quad (\text{Eq. C.14})$$

the equations can be shown to correspond.

The last stage of the f3dm33 subroutine consists of the calculation of the history variables. The hydrostatic pressure and equivalent stress based on sigt are calculated and stored in smcalc and aj1. The difference between the equivalent stress and the current yield stress is stored in dsige. The update of the equivalent plastic strain increment is as follows:

$$depi = \frac{\sigma_{eff} - \sigma_Y}{E_p^a} \quad (\text{Eq. C.15})$$

This is possible because  $H'$  is constant throughout an analysis. The equivalent plastic strain ep is updated with the increment in plastic strain of Eq. 30:

$$\bar{\epsilon}_{t_{n+1}}^p = \bar{\epsilon}_{t_n}^p + \frac{\bar{\sigma} - \sigma_Y}{H'} \quad (\text{Eq. C.16})$$

The value of depi is also stored in epx4 which is grouped in common block /aux14/. To conclude the routine the stress and strain vectors are rotated back to the global axes. For the final rotation a call to subroutine gblm22 is made. This routine transforms the stresses and strains back to the global axes using the inverse of the transformation matrix b that was calculated in the beginning of the routine.

To conclude the routine the stress and strain vectors are rotated back to the global axes. For the final rotation a call to subroutine gblm22 is made. This routine transforms the stresses and strains back to the global axes using the inverse of the transformation matrix b that was calculated in the beginning of the routine.

### **Notes on the plastic stress increment calculation.**

Using an equivalent derivation to Hill [ 44 ] on p333 Eq 18:

$$dW^p = \sigma_{ij} d\epsilon_{ij}^p = \sigma_{ij} : \frac{\partial f}{\partial \sigma_{ij}} d\lambda = n f d\lambda = \sqrt{\frac{2}{3}} \bar{\sigma} d\lambda \quad (\text{Eq. C.17})$$

This is using the fact that  $f$  is homogeneous and defined as follows:

$$f(\sigma_{ij}, \alpha_{ij}, R) = \sqrt{(A : \xi) : \xi} = \sqrt{\frac{2}{3}} \bar{\sigma}$$

which is homogeneous of first order in  $\sigma_{ij}$ , hence:

$$\sigma_{ij} : \frac{\partial f}{\partial \sigma_{ij}} = n f = \sqrt{\frac{2}{3}} \bar{\sigma}$$

Also

$$dW^p = \bar{\sigma} d\bar{\epsilon}^p \quad (\text{Eq. C.18})$$

Combining Equations Eq. C.17 and Eq. C.18 results in:

$$d\lambda = \sqrt{\frac{3}{2}} d\bar{\epsilon}^p$$

Which is a familiar result from Von Mises Plasticity.

Starting from:

$$\begin{aligned} d\phi\sigma &= \frac{1}{H'} A \sigma \\ &= \frac{\bar{\sigma}}{H'} \frac{\partial f}{\partial \sigma} \end{aligned} \quad (\text{Eq. C.19})$$

Using

$$d\bar{\epsilon}^p = \frac{d\bar{\sigma}}{H'}$$

Eq. C.19 can be rewritten as:

$$d\phi\sigma = \frac{\bar{\sigma}}{d\bar{\sigma}} d\bar{\epsilon}^p \frac{\partial f}{\partial \sigma_{ij}} = \frac{\bar{\sigma}}{d\bar{\sigma}} \sqrt{\frac{3}{2}} d\lambda \frac{\partial f}{\partial \sigma_{ij}},$$

or

$$d\mathbf{p} = \frac{\bar{\sigma}}{d\bar{\sigma}} \sqrt{\frac{3}{2}} d\boldsymbol{\varepsilon}^p \quad (\text{Eq. C.20})$$

which is stored in  $dp(1), \dots, dp(6)$ . This tensor is used to calculate  $\bar{dp}$ , which results in  $\bar{dp}$  being:

$$\bar{dp} = \frac{\bar{\sigma}}{d\bar{\sigma}} \sqrt{\frac{3}{2}} d\bar{\varepsilon}^p,$$

which is stored under variable name  $depix$ .

Further down in the routine  $dp(1), \dots, dp(6)$  are modified by multiplying with

$$depi = \frac{d\bar{\sigma}}{H'}$$

and dividing by  $\bar{dp}$ . This results in  $dp(1), \dots, dp(6)$  being equal to the plastic strain increment tensor.

The variables are stored in common /vect15p/, which is not used outside the f3dm33 routine.